

AD-A138 142

EMPLOYMENT OF ADAPTIVE LEARNING TECHNIQUES FOR THE
DISCRIMINATION OF ACQU. (U) GENERAL ELECTRIC CORPORATE
RESEARCH AND DEVELOPMENT SCHENECTA. J W ERKES ET AL.

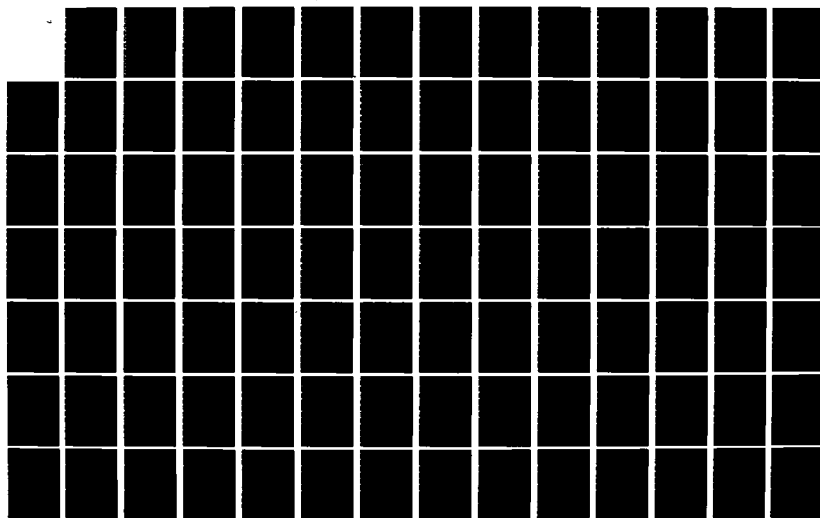
1/4

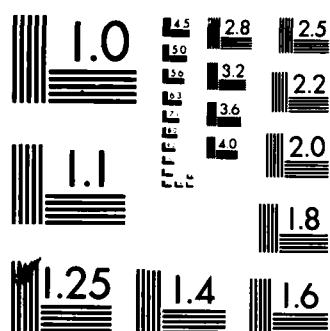
UNCLASSIFIED

NOV 83 83-SRD-060 N00014-82-C-2031

F/G 20/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A 138142

EMPLOYMENT OF ADAPTIVE LEARNING TECHNIQUES FOR THE DISCRIMINATION OF ACOUSTIC EMISSIONS

PHASE I FINAL REPORT
Contract N00014-82-C-2031

November 1983

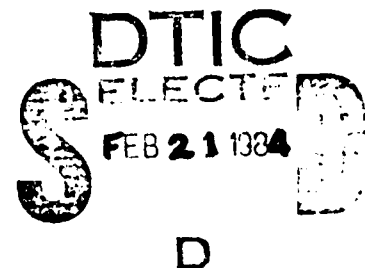
Prepared by

J.W. Erkes, Project Manager
General Electric Company
Corporate Research and Development
Schenectady, New York 12345

Prepared for

H. Chaskelis, Project Manager
Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375

DTIC FILE COPY



83-SRD-060

84 02 17 135

DISTRIBUTION STATEMENT
Approved for public release
Distribution Unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS None		
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Distribution: Distribution Unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 83-SRD-060			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION General Electric Company		6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State and ZIP Code) General Electric Company Corporate Research and Development 1 River Road Schenectady, NY 12345			7b. ADDRESS (City, State and ZIP Code)		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Research Laboratory		8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N0014-82-C-2031		
8c. ADDRESS (City, State and ZIP Code) 4555 Overlook Ave., S.W. Washington, D.C. 20375			10. SOURCE OF FUNDING NOS.		
11. TITLE (Include Security Classification) Employment of Adaptive Learning Techniques for the Discrimination Of Acoustic Emissions (Unclassified)			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
					WORK UNIT NO.
12. PERSONAL AUTHOR(S) Erkes, Joseph W.; McDonald, John F.; Scarton, Henry A.; Tam, Kwok C.; Kraft, Russell P.					
13a. TYPE OF REPORT Phase I Final Report		13b. TIME COVERED FROM 11/81 TO 12/82		14. DATE OF REPORT (Yr., Mo., Day) 11/83	
				15. PAGE COUNT 312	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB. GR.	Acoustic Emission, Digital Signal Processing, Adaptive Learning, Homomorphic Deconvolution, Volterra Filters		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
<p>In this report the work performed during Phase I of the NRL contract on the discrimination of acoustic emissions (AE) is described. The scope of Phase I was limited to three basic areas:</p> <ul style="list-style-type: none">• <i>Analytical Development.</i> The analytical development and assessment of digital signal processing techniques for AE signal dereverberation, noise reduction, and source characterization• <i>Computer Simulation.</i> The modeling and verification of some aspects of key selected techniques through a computer-based simulation• <i>Signal Propagation Characteristics.</i> The study of signal propagation physics and their effect on received signal characteristics for relevant physical situations					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED X SAME AS RPT □ DTIC USERS □				21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Henry Chaskelis			22b. TELEPHONE NUMBER (Include Area Code) (202) 767-3613		22c. OFFICE SYMBOL

DD FORM 1473, 83 APR

EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

Section	Page
1 INTRODUCTION AND SUMMARY	1-1
2 SIGNAL PROCESSING TECHNIQUE	2-1
2.1 Homomorphic Processing	2-1
2.1.1 Amplitude and Arrival Time Correlation	2-1
2.1.2 Pulse Shapes	2-1
2.1.3 Arrival Time Distribution	2-1
2.1.4 Two-Dimensional Statistics	2-1
2.1.5 Phased-Array Imaging	2-2
2.2 Error Propagation	2-2
2.3 Volterra Filtering	2-3
2.4 Pattern Recognition	2-4
3 SIGNAL PROPAGATION PHYSICS	3-1
Appendix	
A Homomorphic Signal Dereverberation for Optimized Phased Array Imaging Systems	Tab A
B Signal Propagation Physics in a Bounded Elastic Medium and the Interpretation of Acoustic Emission Signatures	Tab B

Accession For	
NTI	<input checked="" type="checkbox"/>
PTI	<input type="checkbox"/>
Unprocessed	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A/1	



Section 1

INTRODUCTION AND SUMMARY

SCOPE

In this report the work performed during Phase I of the NRL contract on the discrimination of acoustic emissions (AE) is described. The scope of Phase I was limited to three basic areas:

- *Analytical Development.* The analytical development and assessment of digital signal processing techniques for AE signal dereverberation, noise reduction, and source characterization.
- *Computer Simulation.* The modeling and verification of some aspects of key selected techniques through a computer-based simulation.
- *Signal Propagation Characteristics.* The study of signal propagation physics and their effect on received signal characteristics for relevant physical situations.

The highlights of those results are summarized in the rest of this section, and described in detail in Appendices A and B.

Section 2

SIGNAL PROCESSING TECHNIQUE

The analytical works presented in the proposal are further developed. The theory is generalized to deal with a much wider class of signals. Computer simulations and digital computations are performed to verify the results. The main points are summarized below; details may be found in Appendix A.

2.1 Homomorphic Processing

The basic theory of dereverberation by homomorphic deconvolution is verified. In addition, a number of assumptions made in deriving the basic theory have been removed so that the deconvolution technique is shown to be applicable to a much wider class of signals.

2.1.1 Amplitude and Arrival Time Correlation

The effect of the correlation between the amplitude and the time of arrival of the pulses has been analyzed. It was found that if the amplitude a_i and the arrival time t_i of the pulses are not independent, i.e., $\rho \neq 0$, a nonzero term will be introduced in the cepstrum of the pulses in the region $\tau = 0$, so that the cepstrum will be broadened instead of peaking at the origin $\tau = 0$. This non-zero term is given by

$$g(\tau) + \frac{1}{2} g(\tau) * g(\tau) + \dots$$

where $g(\tau) = \exp(\bar{a}_i \tau / \rho) U(\tau) + \frac{\bar{a}_i}{\rho} \delta(\tau)$, $U(\tau)$ being the unit step function.

2.1.2 Pulse Shapes

The pulse shape is no longer restricted to be Gaussian. It is shown that perfect separation of cepstra is possible for all pulse shapes which are Hermite_∞ expandable. Since Hermite expansions exist for all square-integrable functions where $\int_0^\infty f^2(t) dt$ exists, cepstral deconvolution can be applied to a broad class of pulse shapes.

2.1.3 Arrival Time Distribution

The restriction of the pdf of the arrival time to be either a Gaussian or an ideal impulse is removed. It is shown that for all the pdfs which can be represented by a Graham-Charlies series or an Edgeworth series, the log of the characteristic function is so smooth that its transform inverse produces only singular functions (such as delta, doublet, or triplet functions). Under such circumstances perfect separation of reverberation and signal statistics is possible at least in the continuous domain.

2.1.4 Two-Dimensional Statistics

In a case in which the average of the amplitude of the pulses is zero, the one-dimensional pdf approach is not appropriate. The problem can be restated in terms of its two-dimensional statistics. Analysis indicates that on the assumption that the amplitude and the time of arrival are independent, perfect separation of signal and reverberation using two-dimensional statistics is possible for the general class of pdf and pulse shape mentioned above.

2.1.5 Phased-Array Imaging

The mathematics involved in using a phased array to image multiple (two or three) sources have been worked out. Assuming complete suppression of the side lobes, the closed form solution for the transfer function $H_{ij}(\omega)$ for the path from source j to sensor i is found to be given by

$$H_{ij}(\omega) = \frac{K_i(\omega)}{\sqrt{\sum_i A_i K_i(\omega)}}$$

where

A_i = gain of sensor i

$K_i(\omega)$ = Fourier transform of the dot product between the array pattern and the signal at sensor i with the array "looking" at source j

The incomplete suppression of the side lobes of the phased array gives rise to errors in the calculations of the transfer functions as given above. An iterative method was developed to improve the accuracy of the transfer functions by using each previously calculated estimate of H s to improve the next approximation. The theory behind this procedure is that with multiple, say two, sources in the medium, the small error in measuring the H_{i1} s is a known function of the H_{i2} s and vice versa. The first estimate of the H_{i1} s is plugged into this known function to reduce the error in the H_{i2} approximations. Now with an approximation for the H_{i2} s, the H_{i1} values can be recalculated with a smaller error. This process can be iterated until the changes in H_{i1} and H_{i2} from one iteration to the next are negligibly small.

Once the path transfer functions $H_{ij}(\omega)$ are calculated, deresonation is performed by simply multiplying the signals from the sensors by the inverse of these transfer functions.

2.2 Error Propagation

Fluctuations in moment estimates or other averaging functions that operate on real stochastic data add a certain amount of noise to the results. The tradeoffs between the number of data records averaged and the uncertainty in the resulting cepstra where reverberation must be discerned and separated have been analyzed. With the simplification that the average of the Fourier transform of the pulses $\overline{H_{pi}(\omega)} = 1$, and the estimator $\overline{Y(\omega)}$ (average of the Fourier transform of the measured waveforms) is unbiased, the error term due to the noise $n(t)$ is given by:

$$\Delta \left[\log \frac{\hat{Y}(\omega)}{\overline{Y(\omega)}} \right] = \frac{\Delta \left[\frac{\hat{N}(\omega)}{\overline{Y(\omega)}} \right]}{\overline{Y(\omega)}} + \frac{\Delta \left[\sum_i a_i e^{-j\omega\tau_i} \right]}{\sum_i a_i e^{-j\omega\tau_i}}$$

where $N(\omega)$ is the Fourier transform of $n(t)$. Relaxing the restriction that $H_{pi}(\omega) = 1$ will only increase the error slightly. The term affected is the last one in the above equation.

Noise in the data also affects the phase unwrapping routine in computing the cepstrum. The phase noise could induce a discontinuity in the unwrapped phase of 2π and cause an erroneous unwrapped phase to be computed.

In this project an upper bound on the probability of an incorrect phase and cepstrum given an amount of random phase noise due to measurement and numerical

roundoff error was derived, which indicates the amount of averaging on the ensemble records and the necessary length T of each record to achieve that bound. Let $x(t)$ and $y(t)$ be two stationary random processes representing the lumped phase noise. The probability of a cepstral error is given by $Q(\kappa)$ where $Q(\cdot)$ is the area under the Gaussian tail, $\kappa = R_{xy}(\tau)/\sigma_{xy}(\tau, T)$, $R_{xy}(\tau)$ being the cross-correlation of the random processes representing the lumped phase noise, and $\sigma_{xy}(\tau, T)$ is given by

$$\sigma_{xy}^2(\tau, T) = \frac{2}{T^2} \int_0^T (T-\nu) [P_{xy}^2(\tau, \nu) - R_{xy}^2(\tau)] d\nu$$

$$P_{xy}^2(\tau, u-t) = \langle x(t)y(t+\tau)x(u)y(u+\tau) \rangle$$

$\sigma_{xy}(\tau, T)$ also puts a limit on the deviation of every point on the unwrapped phase from its true value and can be used to limit the phase noise for cases where no erroneous jogs are present from improperly selected ambiguities.

2.3 Volterra Filtering

Nonlinear filters were developed to suppress background noise in the raw data $y(t)$ to produce a signal $\hat{x}(t)$ which is as close as possible to the desired impulse signal $x(t)$. The equations governing the design of an optimal quadratic nonlinear least mean square estimation filter (QLMSE filter) with kernels $h_1^0(\tau)$ and $h_2^0(\tau)$ were derived. The formulation is based on a truncated Volterra series, and resulted in the following pair of integral equations:

$$\int_{-\infty}^{\infty} h_1^0(\tau) = R_{yyy}^{[3]}(\tau, \tau_1, \tau_2) d\tau + \int \int_{-\infty}^{\infty} h_2^0(\tau_1, \tau_2')$$

$$R_{yyyy}^{[4]}(\tau_1, \tau_2, \tau_1', \tau_2') d\tau_1' d\tau_2' = R_{xyy}^{[3]}(0, \tau_1, \tau_2)$$

$$\text{for } \tau_1 \geq 0, \tau_2 \geq 0$$

where

$$R_{yyy}^{[3]}(\tau, \tau_1, \tau_2) = \overline{y(t-\tau) y(t-\tau_1) y(t-\tau_2)}$$

$$R_{yyyy}^{[4]}(\tau_1, \tau_2, \tau_1', \tau_2') = \overline{y(t-\tau_1) y(t-\tau_2) y(t-\tau_1') y(t-\tau_2')}$$

$$R_{xyy}^{[3]}(0, \tau_1, \tau_2) = \overline{x(t) y(t-\tau_1) y(t-\tau_2)}$$

For digital computations the discrete version of the optimal Volterra filters must be used. By the use of a multidimensional Volterra expansion, one can write the most general nonlinear recursive IIR quadratic filter as

$$y_i = \sum_{j=0}^N h_j^{(1)} x_{i-j} + \sum_{k=1}^L g_k^{(1)} y_{i-k} \\ + \sum_{j_1=0}^{L_1} \sum_{j_2=0}^{L_2} l_{j_1, j_2}^{(2)} x_{i-j_1} x_{i-j_2}$$

$$\begin{aligned}
& + \sum_{k_1=1}^{M_1} \sum_{k_2=1}^{M_2} m_{k_1, k_2}^{(2)} y_{i-k_1} y_{i-k_2} \\
& + \sum_{p_1=0}^{N_1} \sum_{q_2=1}^{N_2} n_{p_1, q_2}^{(2)} x_{i-p_1} y_{i-q_2}
\end{aligned}$$

The quadratic terms contain feedforward, feedback, and feedforward-feedback expressions, respectively. The errors due to roundoff and coefficient quantization can be estimated using methods similar to the one employed by Liu for linear systems.

An adaptative method was also devised for determining an optimum Volterra predictor. The procedure is to adjust a number of kernels adaptively to minimize the difference between the output of the predictor and the confirmed value of the measurement at a later time.

2.4 Pattern Recognition

The pulse shape features, amplitude statistics, and event statistics are used to characterize the event that generates the pulses.

To represent the continuous waveforms with a discrete number of coefficients which can be ranked-ordered according to their importance in representing these waveforms, the Karhunen-Loeve expansion is used. This expansion is an orthogonal series which has the property that the coefficients of expansion are uncorrelated. In this way the basis functions provide a maximal information content about the pulses.

The Karhunen-Loeve basis function for each kind of event which generates the waveforms $x(t)$ are obtained by solving the following eigenvalue problem,

$$\lambda_j \phi_j(t) = \int_0^T K_x(t, u) \phi_j(u) du$$

where

$$K_x(t, u) = E \left\{ [x(t) - \bar{x}(t)] [x(u) - \bar{x}(u)] \right\}$$

To characterize each pulse, its expansion coefficients in the basis functions for each kind of event are computed. The "distances" between these coefficients and those of the different the basis of the closest "distance" using Bayes rule.

Section 3

SIGNAL PROPAGATION PHYSICS

A brief overview of the signal propagation effort is given below; details may be found in Appendix B.

In order to provide some guidance for the subsequent experimental verification of the results of Phase I, an effort was made to understand and characterize the signal propagation physics under physically representative conditions. That effort provided the basis for the experimental plan subsequently executed under Phase II.

In general, the specimen ringdown transfer function depends in a very complicated way on many factors: the nature of the exciting signal, the geometry and material characteristics of the specimen, and the direction and location of the transducer. The physics of wave propagation is analyzed to develop insights on these phenomena. Two methods of analysis, the modal analysis and the generalized ray theory, are described and employed for such purpose.

Appendix A

Appendix A

**HOMOMORPHIC SIGNAL DEREVERBERATION FOR OPTIMIZED
PHASED ARRAY IMAGING SYSTEMS**

Russell P. Kraft

**A Thesis Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute
in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY
in Electrical Engineering**

TABLE OF CONTENTS

	Page
LIST OF TABLES	iv
LIST OF FIGURES.	v
ACKNOWLEDGEMENT.	x
ABSTRACT	xi
1. INTRODUCTION AND HISTORICAL REVIEW	1
2. SINGLE SOURCE DEREVERBERATION.	6
2.1 Introduction to Homomorphic System of Convolution.	6
2.2 The Dereverberation Problem Formulation.	9
2.3 Signal Dereverberation by use of Ad Hoc One-Dimensional Probability Distribution Functions	11
2.4 Signal Dereverberation through Homomorphic Filtering with One-Dimensional Statistics.	13
2.5 Signal Dereverberation for Source with Dependence Between Amplitude and Arrival Time	16
2.6 Signal Dereverberation by use of Ad Hoc Two- Dimensional Probability Distribution Functions	21
2.7 Signal Dereverberation through Homomorphic Filtering with Two-Dimensional Statistics.	25
2.8 Treatment of Hermite Expansion Source Pulse Case	27
2.9 Errors Due to General Source Pulse Transfer Function	31
2.10 Signal Dereverberation Within a Dispersive Channel	33
2.11 Numerical Considerations of the IDFT of $(j\omega)^i/i!$	34
2.12 Programming Considerations and Simulations	45
2.13 Relations Between Cepstrum and Predictor Coefficients.	66
2.14 Errors Due to Estimation Noise	72
3. DUAL SOURCE DEREVERBERATION.	78
3.1 The Dual Source Problem Formulation.	78
3.2 Solution to Dual Source Nonlinear Equations.	88
3.3 Improving the Accuracy of H_{i1} and H_{i2} and Image Enhancement.	89
3.4 Convergence of Iterative Solution of H_{i1} and H_{i2}	92
3.5 Introduction to the Three Source Problem	94
3.6 Array Dereverberation Simulations.	96
4. ARRAY OPTIMIZATION	115
4.1 Phased Array Fundamentals.	115
4.2 Phased Array Diffraction Suboptimization	121

	Page
4.3 Diffraction Pattern Simulations of Phased Arrays.	126
4.4 Grating Lobe Suppression by Nonuniform Spacing.	145
4.5 Minimax Beam Pattern Optimization of Nonuniformly Spaced Cartesian Arrays.	156
4.6 Minimax Beam Pattern Optimization of Fresnel Lens Arrays.	159
4.7 Minimax Array Optimization Observations	174
5. DISCUSSION AND CONCLUSIONS.	177
6. LITERATURE CITED.	179
7. APPENDICES.	186
7.1 $F^{-1}[(j\omega)^i/i!]$ Simulation Program	187
7.2 Phase Unwrapping and Cepstrum Computation Program	192
7.3 Iterative Solution of Channel Response Program.	219
7.4 Dual Source Imaging Program	223
7.5 Fresnel Diffraction Array Simulation Program.	227
7.6 Minimax Array Optimization Program.	232
7.7 One Dimensional Graphics Plotting Program	245
7.8 Two Dimensional Surface Plotting Program.	251
7.9 Example Dereverberating a Lamb Wave Convolved With a Resonant Sensor	268

LIST OF TABLES

	Page
Table 2.3.1 Common Probability Distribution Functions and Their Corresponding Characteristic Functions	12
Table 2.6.1 Ad Hoc Functions for $Q(\omega)Q'(\omega')$	22
Table 2.6.2 Ad Hoc Functions for $Q(\omega)Q'(\omega')$ when $\omega = \omega'$	24

LIST OF FIGURES

	Page
Figure 2.1.1 Homomorphic System of Convolution.	7
Figure 2.1.2 Characteristic Homomorphic System of Convolution And Deconvolution.	8
Figure 2.11.1 Magnitude of Inverse Fourier Transform of $j\omega$; 256 points	36
Figure 2.11.2 Phase of Inverse Fourier Transform of $j\omega$; 256 points	37
Figure 2.11.3 Magnitude of Inverse Fourier Transform of $j\omega$; 2048 points.	39
Figure 2.11.4 Phase of Inverse Fourier Transform of $j\omega$; 2048 points.	40
Figure 2.11.5 Magnitude of Inverse Fourier Transforms of $(j\omega)^5/5!$.	41
Figure 2.11.6 Magnitude of Inverse Fourier Transform of $(j\omega)^9/9!$.	42
Figure 2.11.7 Magnitude of Inverse Fourier Transform of $(j\omega)$, $\frac{(j\omega)^5}{5!}$ and $\frac{(j\omega)^9}{9!}$	43
Figure 2.11.8 Log of Magnitude of Inverse Fourier Transform of $(j\omega)$, $\frac{(j\omega)^5}{5!}$ and $\frac{(j\omega)^9}{9!}$	44
Figure 2.12.1 Noise Pulse, $x(t) = e^{- t /8}$; 512 points.	48
Figure 2.12.2 Fourier Transform of Noise Pulse, $F[e^{- t /8}]$; 512 points	49
Figure 2.12.3 Cepstrum of Noise Pulse; 512 points	50
Figure 2.12.4 Noise Pulse, $x(t) = e^{- t /8}$; 2048 points	51
Figure 2.12.5 Fourier Transform of Noise Pulse, $F[e^{- t /8}]$; 2048 points	52
Figure 2.12.6 Cepstrum of Noise Pulse; 2048 points	53
Figure 2.12.7 Delayed Noise Pulse $e^{- t-64 /8}$; 2048 points	54
Figure 2.12.8 Blow-up of Noise Pulse of 2.12.7	55

	Page
Figure 2.12.9 Fourier Transform of Delayed Noise Pulse; 2048 points	56
Figure 2.12.10 Cepstrum of Delayed Noise Pulse; 2048 points. . . .	57
Figure 2.12.11 Blow-up of Cepstrum of 2.12.10.	58
Figure 2.12.12 Dual Noise Pulses, $e^{- t-32 /8} + .5 e^{- t-160 /8}$; 2048 points	59
Figure 2.12.13 Fourier Transform of Dual Noise Pulses; 2048 points	60
Figure 2.12.14 Cepstrum of Dual Noise Pulses; 2048 points	61
Figure 2.12.15 Normalized Deresonated Noise Pulse (Magnitude); 2048 points	62
Figure 2.13.1 Adaptive Linear Predictor of Source Signal.	69
Figure 3.1.1 Signal Path Delays from Points P_1 and P_2 in a Medium to N Sensor Signals $y_i(t)$	79
Figure 3.1.2 Path Impulse Functions from Sources $v_1(t)$ and $v_2(t)$ to N Sensor Signals $y_i(t)$	81
Figure 3.6.1 Plot of Position of Sources (Tall Peaks), Sensors (Short Peaks) and Reflective Boundary	101
Figure 3.6.2 Image of Left Source Activated alone Without Echo .	102
Figure 3.6.3 Image of Right Source Activated alone Without Echo. .	103
Figure 3.6.4 Image of Left Source Activated alone With Echo. . .	104
Figure 3.6.5 Image of Right Source Activated alone With Echo. . .	105
Figure 3.6.6 Image of Both Sources Activated Simultaneously With Echos	106
Figure 3.6.7 Image of Both Sources Activated Simultaneously Without Echos (Ideal Dereverberation Case).	107
Figure 3.6.8 Dereverberated Image of Figure 3.6.6.	108
Figure 3.6.9 Comparison of Ideal Echoless Image (top) and Dereverberated Image (bottom)	109
Figure 3.6.10 Intermediate Dereverberation Result of Figure 3.6.8	110

	Page
Figure 3.6.11	Ideal Channel Transfer Functions $H_{11}(\omega)$ (top) and $H_{21}(\omega)$ (bottom) 111
Figure 3.6.12	First Approximation of Channel Transfer Functions $H_{11}(\omega)$ (top) and $H_{21}(\omega)$ (bottom). 112
Figure 3.6.13	Approximation of Channel Transfer Functions $H_{11}(\omega)$ (top) and $H_{21}(\omega)$ (bottom) after 5 Iterations. 113
Figure 3.6.14	Approximation of Channel Transfer Function $H_{11}(\omega)$ (top) and $H_{21}(\omega)$ (bottom) after 20 Iterations. 114
Figure 4.2.1	Beam Pattern; 16 Element Linear Array, Kaiser Gains, Far Field. 127
Figure 4.2.2	Beam Pattern; 16x16 Array, Kaiser Gains, Far Field mapped by McClellan Transformation. 128
Figure 4.3.1	Coordinate System for Fresnel Diffraction Pattern . 129
Figure 4.3.2	Beam Pattern; Uniform Gains, Unfocused. 132
Figure 4.3.3	Beam Pattern; Uniform Gains, Unfocused, Steered . . 132
Figure 4.3.4	Beam Pattern; Kaiser Gains, Unfocused 133
Figure 4.3.5	Beam Pattern; Kaiser Gains, Unfocused, Steered. . . 133
Figure 4.3.6	Beam Pattern; Kaiser Gains, Focused 135
Figure 4.3.7	Beam Pattern; Kaiser Gains, Focused, Steered. . . . 135
Figure 4.3.8	Fresnel Diffraction Pattern for a 5x5 Array in the Very Near Field 137
Figure 4.3.9	Beam Pattern; 5x5 Array, Uniform Gains, Unfocused, $z=.8m$ 138
Figure 4.3.10	Beam Pattern; 5x5 Array, Uniform Gains, Unfocused, Steered 3° , $z=.8m$ 140
Figure 4.3.11	Beam Pattern; 5x5 Array, Uniform Gains, Unfocused, $z=.125m$ 141
Figure 4.3.12	Beam Pattern; 5x5 Array, Uniform Gains, Focused, $z=.125m$ 142

	Page
Figure 4.3.13	Beam Pattern; 5x5 Array, Uniform Gains, Unfocused, Steered 2.5°, z=.125m 143
Figure 4.3.14	Beam Pattern; 5x5 Array, Uniform Gains, Focused, Steered 2.5°, z=.125m 144
Figure 4.3.15	Beam Pattern; 5x5 Array, Kaiser Gains, Unfocused, z=.125m 146
Figure 4.3.16	Beam Pattern; 5x5 Array, Kaiser Gains, Focused, z=.125m 147
Figure 4.3.17	Beam Pattern; 5x5 Array, Kaiser Gains, Focused, Steered 2.5°, z=.125m 148
Figure 4.4.1	Beam Pattern; 11x11 Array, Uniform Gains, Focused, Uniform Spacings. 151
Figure 4.4.2	Beam Pattern; 11x11 Array, Uniform Gains, Focused, Elimination of Multiple Spacings. 152
Figure 4.4.3	Planar Array; 11x11 Elements, Elimination of Multiple Spacings 153
Figure 4.4.4	Planar Array; 11x11 Elements, Arithmetic Progression Spacings. 154
Figure 4.4.5	Planar Array; 25x25 Elements, Logarithmic Spacings. 155
Figure 4.5.1	Beam Pattern; 16 Element Linear Array, Uniform Gains, Focused, Uniform Spacings. 160
Figure 4.5.2	Beam Pattern; 16 Element Linear Array, Optimized Gains, Phases and Spacings. 161
Figure 4.5.3	Beam Pattern; 11x11 Array, Uniform Gains, Focused, Uniform Spacings. 162
Figure 4.5.4	Beam Pattern; 11x11 Array, Optimized Gains, Phases and Spacings. 163
Figure 4.6.1	Fresnel Zone Plate Diffraction Pattern in the Principal Focal Plane 165
Figure 4.6.2	Fresnel Lens Array Axial Beam Pattern Showing False Focal Spots 166

	Page
Figure 4.6.3	Fresnel Lens Array Axial Beam Pattern After Optimization 166
Figure 4.6.4	Cross Section of the Beam Pattern at the Focal Plane of an 11 Element Fresnel Lens Array. 170
Figure 4.6.5	Cross Section of the Beam Pattern at the Focal Plane of an Optimized 11 Element Fresnel Lens Array. 171
Figure 4.6.6	Beam Pattern of an 11 Element Fresnel Lens Array . 172
Figure 4.6.7	Beam Pattern of an Optimized 11 Element Fresnel Lens Array 173
Figure 4.7.1	Optimized 11 Element Array Dereverberation Simu- lation 176
Figure 7.9.1	Lamb Wave (solid line) and Resonant Sensor Impulse Response (dashed line) 272
Figure 7.9.2	Convolved Lamb Wave and Sensor Response. 273
Figure 7.9.3	Fourier Transform of Convolved Lamb Wave and Sensor Response 274
Figure 7.9.4	Cepstrum of Convolved Lamb Wave and Sensor Response 275
Figure 7.9.5	Dereverberated Lamb Wave 276
Figure 7.9.6	Sensor Time Signal Remaining After Filtering Out Lamb Wave. 277

ACKNOWLEDGEMENT

The author would like to express his gratitude to Dr. John F. McDonald for his guidance, encouragement and perseverance during the tenure of this research project. Also, special thanks go to Dr. Howard Kaufman, Dr. Pankaj K. Das, and Dr. Henry A. Scarton for their help and contributions to this project. Also, the author is deeply indebted to Dr. Frederick Ahlgren and Dr. Joseph Erkes of General Electric Corporation, and Dr. Henry Chaskelis of Naval Research Laboratory for their financial support of this research endeavor.

Finally, an especially sincere thank you is extended to his wife for her patience, constant support and unfaltering faith during the preparation of this manuscript.

ABSTRACT

The general discipline of phased array imaging presents a variety of problems which depend on the specific application. To date, phased array antennas have been used successfully for imaging in audio, ultrasonic and radar frequency ranges. Some of the problems that exist in these various disciplines are the presence of grating lobes and aperture side lobes, lack of depth resolution when imaging in the far field, narrow elemental angular beam-width, poor interelement isolation, and in some cases the complete loss of the image due to the presence of multiple echos in the signals received by the array elements.

Grating lobes can be eliminated by reducing the array element spacing to less than one half the wavelength of the travelling wave in the material being imaged. However, in designing multi-element phased arrays it is always desirable to keep the number of elements in the array to a minimum. This results in economic savings in the construction of the array and in computational savings in the processing of the collected data for the reconstruction of images. With this reduction in the number of elements comes an immediate trade-off in the array's performance between image resolution and the introduction of ambiguities into the image through the grating lobes. The negative effects of grating and side lobes and also the poor resolution in the near field can be reduced by optimally adjusting the interelement spacings, shading the element gains and varying the element phases of the array. The array element signals

which contain echos (a problem arising in the nondestructive testing of solids) can be dereverberated through the use of homomorphic filtering before an image is made. This method works well when certain conditions are met.

When all these methods are applied simultaneously, significant improvements in the resolution of phased array images can be observed. This work examines the problems listed here and ascertains viable solutions which are demonstrated through simulations.

PART 1

INTRODUCTION AND HISTORICAL REVIEW

In many applications of sonic imaging echos resulting from wall reflection and resonance due to internal wall reverberating transmission are significant sources of distortion. These echos and reverberations produce many false target images which are usually indistinguishable from true target images using simple ad hoc correlation techniques. Of course, propagation through a linear reverberant medium may be modeled by a transfer function or linear filter for which an inverse exists. Unfortunately, this filter is not known a priori and it must be estimated from stochastic data.

The purpose of this study is to devise a scheme to deresonate the signals received at the individual elements using some information about the source signal itself and its power spectrum. This will be accomplished with the use of homomorphic deconvolution. In addition, techniques will be investigated to remove false signal sources due to echo reverberations in the presence of two or more real signal sources.

Also, a set of algorithms and programs will be presented for the implementation of the signal processing routines developed in this study for their use on a host computer to determine the locations of the sources. A complete set of simulations is included which test out the algorithms and programs and extensively use computer graphics to present the results of these simulations.

This study represents a continuation of earlier analysis of phased array processing techniques for sonic imaging of passively emitted signals performed in 1979. In the earlier study [1] analysis was done on degradation and maximum error of the signal due to processing on less than the full set of cross correlation terms. Also studied was the optimization of the array by adjustment of the gains, positions and individual phases of the array elements. The results of this earlier study are shown to be extremely relevant to homomorphic array processing. The final results of this thesis supplement, extend, and summarize the improvement possible in array images by applying these optimization techniques. Along with the theory of array optimization are presented results of many simulations of optimized one and two dimensional arrays also with extensive use of computer graphics.

Ultrasonic and audio imaging using multi-element phased array antennas encounter many problems which are not present in their radar and sonar counterparts. In addition to the ambiguities which arise from grating and aperture sidelobes (inherent in arrays designed to reduce the total number of elements for material and processing-time savings) the utilization of such arrays in noninfinite, bounded volumes gives rise to ambiguities due to echos reflected from the container boundaries. Hence, the source signals are contaminated with reverberation noise. Added to this are other ambiguities arising from false foci which appear when two or more coherent sources are present, using accumulative correlation array processing techniques

(ACAP) [1]. Concentrating on the signal reverberation problem and the multiple source problem, methods for resolving these ambiguities will be found using only certain available information about the general statistics of the signals, little or no knowledge of the geometry of the containers and homomorphic processing.

To date, the literature is filled with articles on the many varied aspects under the general topic of phased array imaging. They include design of individual piezoelectric elements and arrays, design of hybrid electronic chips to drive arrays, processing algorithms of special array configurations, performance of radar and waveguide arrays, reduction of errors in bandlimited arrays, effects of quantization error in array image processing, and analysis of array applications in tomography and holography. These problem areas, although extremely important, are not the focus of this thesis and will not be dealt with further than mentioned here, and no attempt will be made to even begin to cite any of their numerous references. A historical review will follow the developments in homomorphic and cepstral processing and array optimization with their immediate general signal processing precedents.

Starting in 1965, some analysis was done [21,22] on the distortion and added echos of signals reflected off lossy layers. Around the same time, work done on speech analysis and synthesis [13,15,16] revealed that logging the spectrum and inverting allowed for the analysis of echos in signals and led to the birth of the cepstrum. This found useful application in speech, seismic and

nondestructive testing processing [14,17,18,19,20] which led to the developments in sections 2 and 3 in this work.

Array imaging necessary in those two sections led to the analysis and design of optimized arrays for imaging. Early work in improved array performance began around 1960 with ad hoc adjustments of element spacings and gains separately [30,40,66,67].

Noting the similarities between optimal array design and optimal filter design permitted the use of the many results of one and two dimensional optimal filter design algorithms [23,24,25,26,60,62,70] for the use in optimal array designs. Simultaneously, work was being done to find efficient methods of processing array information [27,28,29]. Direct design of optimal and suboptimal arrays had begun for the minimization of grating lobes and/or side lobes through gain or spacing adjustments [21,32,33,34,38] and several generalized routines [39,43,44,45] could be adapted to the array optimization problem with very good results. Some improvement in array response could be gained through the use of acoustic lenses [35] but this has many drawbacks. A natural extension of Cartesian arrays for ultrasonic imaging led to a study of Fresnel lens arrays [36,37,73,78] for subsequent optimization.

This work taps the significant results in all these areas and extends them for near field imaging with simultaneous optimization of gains, phases, and spacings and cepstral processing for source imaging with the removal of false targets. The original contributions of this thesis include the application of cepstral processing

to random pulse trains and the development of cepstral array processing in Chapter 2, after the introduction of the homomorphic system, and the complete development of the cepstral processing for the dereverberation of dual sources in Chapter 3. Previously developed material in 2.1, 2.3, 2.6, 2.12, and 2.13 was extended and adopted for this dereverberation application. In the array optimization area the original contributions are the extension of the design of suboptimized arrays through linear filter design techniques into the near field and the minimax optimization of grating and side lobes by simultaneous adjustment of gains, spacings and phases for any field of view of arrays.

PART 2

SINGLE SOURCE DEREVERBERATION

2.1 Introduction to Homomorphic Systems of Convolution

A linear system is one that is defined under the operation of addition so that if $x(n)$ is the input and $y(n) = L[x(n)]$ is the output then if $x_1(n) + x_2(n)$ or $ax(n)$ are the inputs, the outputs respectively are $L[x_1(n)] + L[x_2(n)]$ and $aL[x(n)]$. A conventional homomorphic system [2] of convolution is likewise defined for convolution so that for $x(n)$ being the input yields output $y(n) = H[x(n)]$ and input $x_1(n)*x_2(n)$ gives $H[x_1(n)]*H[x_2(n)]$. This system can be realized by three stages. The first maps convolution to addition; the second, being a linear system, maps addition to addition; and the third, which maps addition back to convolution. The above three blocks in the frequency domain would map multiplication to addition, addition to addition, and addition to multiplication respectively. The first block in the time domain could be further broken down into three functions, ones that map convolution to multiplication, multiplication to addition, and addition to addition. These three functions are a Fourier transform, log and inverse Fourier transform. The third block in the time domain system undoes the first block, mapping addition to addition, addition to multiplication, and multiplication to convolution with the three functions Fourier transform, exponential, and inverse Fourier transform. These functions and blocks, along with their equivalent input and output equations, are illustrated in the following figures 2.1.1 and 2.1.2.

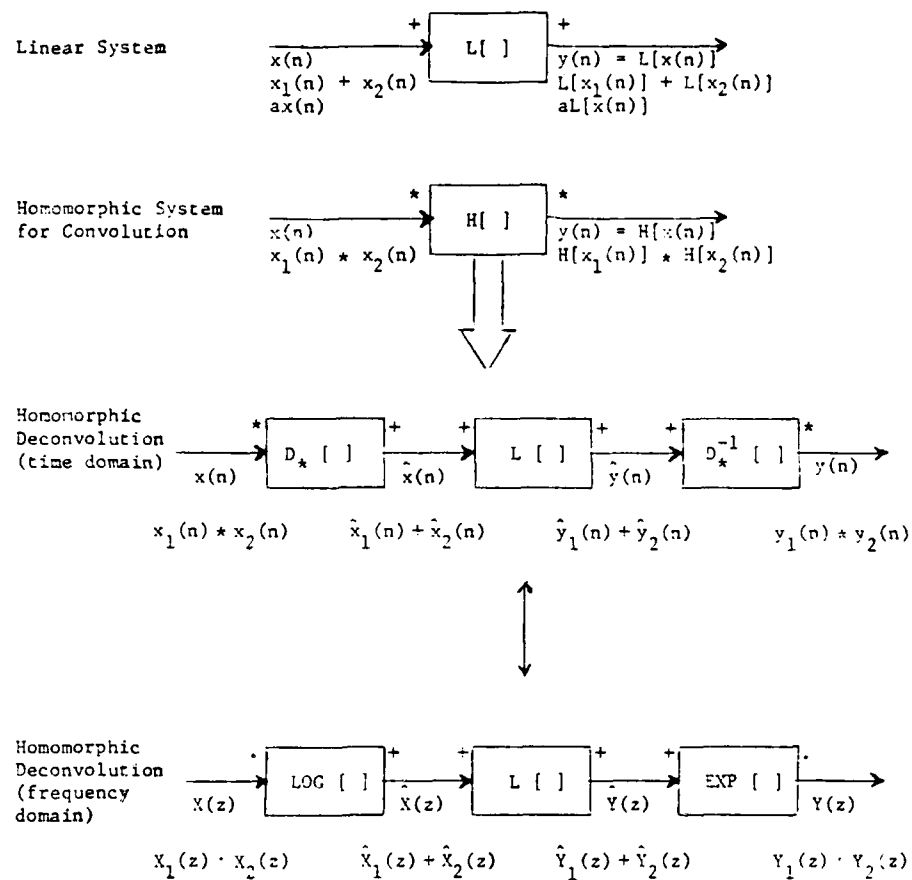


Figure 2.1.1 Homomorphic System of Convolution

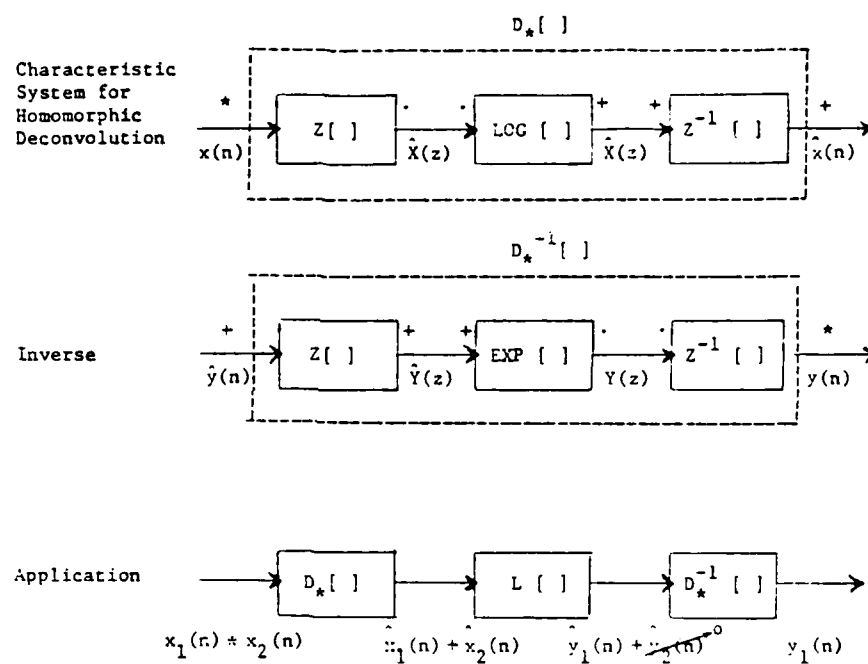


Figure 2.1.2 Characteristic Homomorphic System of Convolution and Deconvolution

As an application, if a signal received is the convolution of a desired signal and a channel transfer function, the desired signal can be recovered by passing the receiver signal through a homomorphic filter which results in the channel function additive in time and hopefully disjoint with the desired signal so that it can be masked out with only the desired signal remaining.

2.2 The Dereverberation Problem Formulation

For various reasons, a direct application of the results of the previous section will not be directly applicable to stochastic dereverberation. The general problem of signal dereverberation can be stated as trying to find the transfer function of the signal path between the source and the receiver. If $H_r(\omega)$ is the transfer function of this path, $Y(\omega)$ the received signal and $X(\omega)$ the source, this can be written in the Fourier transform domain as

$$Y(\omega) = H_r(\omega) X(\omega) \quad (2.2.1)$$

Assuming $X(\omega)$ is a series of gated impulses in time of amplitude a_i with $G(\omega)$ being the transformed gating function $g(t)$, then

$$X(\omega) = \left(\sum_i a_i e^{-j\omega\tau_i} \right) * G(\omega) \quad (2.2.2)$$

For the moment, it will be assumed that $g(t)$ is a constant unity for all time and $G(\omega)$ can therefore be dropped out of the expression since convolving with a delta function returns the original function. Thus,

$$Y(\omega) = H_r(\omega) \cdot \left[\sum_i a_i e^{-j\omega\tau_i} \right] \quad (2.2.3)$$

is obtained from (2.2.1) and (2.2.2) and the aforementioned assumption.

Since $X(\omega)$ is a random signal, time averaging must be utilized to obtain reliable values of $H_r(\omega)$. Therefore

$$\overline{Y(\omega)} = H_r(\omega) \cdot \overline{\left[\sum_i a_i e^{-j\omega\tau_i} \right]} \quad (2.2.4)$$

Taking the averaging inside the summation gives

$$\overline{Y(\omega)} = H_r(\omega) \cdot \left[\sum_i \overline{a_i e^{-j\omega\tau_i}} \right] \quad (2.2.5)$$

and assuming independence between a_i and τ_i , the amplitude and delay time of the impulse respectively gives

$$\overline{Y(\omega)} = H_r(\omega) \cdot \left[\sum_i \overline{a_i} \overline{e^{-j\omega\tau_i}} \right] \quad (2.2.6)$$

$$= H_r(\omega) \cdot \left[\sum_i \overline{a_i} Q(\omega) \right] \quad (2.2.7)$$

$$= H_r(\omega) \cdot Q(\omega) \cdot \sum_i \overline{a_i} \quad (2.2.8)$$

where $Q(\omega)$ is the characteristic function defined as [12]

$$Q(\omega) = \int_{-\infty}^{\infty} e^{-j\omega\tau} p_{\tau}(\tau) d\tau \quad (2.2.9)$$

of the probability distribution function (pdf) of the impulse's time delay, which can be brought outside the summation. This can be rewritten in terms of $H_r(\omega)$ as

$$H_r(\omega) = \frac{\overline{Y(\omega)}}{Q(\omega) \sum_i \overline{a_i}} \quad (2.2.10)$$

$H_r(\omega)$ can then be found if the statistical distribution of the τ_i 's and the average amplitude of the impulses, $\overline{a_i}$, are known and provided that $\overline{a_i}$ is non-zero.

$Y(\omega)$ is the average at a particular frequency over several separate measurement records. Since $Y(\omega)$ is measured the only weak assumption is that $Q(\omega)$ is known.

2.3 Signal Dereverberation by Use of Ad Hoc One-Dimensional Probability Distribution Functions

The problem of finding $Q(\omega)$ in (2.2.10) can be resolved in two ways. The first would be to take many measurements on τ_i (if these can be measured directly) and plot the resultant pdf. From this $Q(\omega)$ can be calculated by taking its Fourier transform.

The second method would be to take a few measurements on τ_i and then make a best fit approximation of its pdf to a common ad hoc density function. The characteristic function of the pdf would then also be known, except for the scaling constants which would need to be calculated and substituted into the equation.

A list of these common pdf's $p(\tau)$, follows in Table 2.3.1 with their corresponding characteristic functions $Q(\omega)$. These probability functions were found in [12] and are general functions that can be utilized in this application.

<u>Name</u>	<u>pdf, $p(\tau)$</u>	<u>Characteristic function, $Q(\omega)$</u>
Uniform	$\frac{1}{2b} \Pi\left(\frac{\tau - c}{2b}\right)$	$e^{jc\omega} \frac{\sin b\omega}{b\omega}$
Triangle	$\frac{1}{2b} \Lambda\left(\frac{\tau - 2c}{2b}\right)$	$e^{j2c\omega} \frac{\sin^2 b\omega}{b^2 \omega^2}$
Normal	$\frac{1}{\sigma\sqrt{2\pi}} e^{-(\tau - \bar{\tau})^2 / 2\sigma^2}$	$e^{(j\bar{\tau}\omega - \sigma^2\omega^2/2)}$
Cauchy	$\frac{\alpha/\pi}{\alpha^2 + \tau^2}$	$e^{-\alpha \omega }$
Laplace	$\frac{\alpha}{2} e^{-\alpha \tau }$	$\frac{\alpha^2}{\alpha^2 + \omega^2}$

TABLE 2.3.1 Common Probability Distribution Functions and Their Corresponding Characteristic Functions [12]

2.4 Signal Dereverberation Through Homomorphic Filtering with One-Dimensional Statistics

Equation (2.2.8) can also be solved for $H_r(\omega)$ through the use of homomorphic filtering. This can be accomplished by taking the natural log of both sides to obtain

$$\log \overline{Y(\omega)} = \log H_r(\omega) + \log Q(\omega) + \log \left[\sum_i \overline{a_i} \right] \quad (2.4.1)$$

If the pdf of τ_i is Gaussian, then

$$\log Q(\omega) = m \cdot j\omega + \frac{\sigma^2}{2} (j\omega)^2 \quad (2.4.2)$$

and

$$\log \overline{Y(\omega)} = \log H_r(\omega) + m \cdot j\omega + \frac{\sigma^2}{2} (j\omega)^2 + \log \left[\sum_i \overline{a_i} \right] \quad (2.4.3)$$

Taking the cepstrum (inverse transform of the log of the transform) yields

$$\begin{aligned} \int_{-\infty}^{\infty} \log \overline{Y(\omega)} e^{j\omega\tau} d\omega &= \int_{-\infty}^{\infty} \log H_r(\omega) e^{j\omega\tau} d\omega + \int_{-\infty}^{\infty} m j\omega e^{j\omega\tau} d\omega + \int_{-\infty}^{\infty} \frac{\sigma^2}{2} (j\omega)^2 e^{j\omega\tau} d\omega \\ &\quad + \int_{-\infty}^{\infty} \log \left[\sum_i \overline{a_i} \right] e^{j\omega\tau} d\omega \end{aligned} \quad (2.4.4)$$

If $H_r(\tau)$ is the inverse transform of $\log H_r(\omega)$ we obtain:

$$y(\tau) = H_r(\tau) + m \frac{d}{d\tau} \delta(\tau) + \frac{\sigma^2}{2} \frac{d^2}{d\tau^2} \delta(\tau) + \log \left[\sum_i \overline{a_i} \right] \delta(\tau)$$

(2.4.5)

Evidently, the log of Q in the Gaussian case is extraordinarily smooth except in the vicinity of its D.C. frequency components which result in the delta and doublet functions of equation (2.4.5).

For $\tau \neq 0$ this can be simplified to

$$Y(\tau) = H_r(\tau) \quad ; \tau > 0 \quad (2.4.6)$$

That is except at $\tau = 0$ the reverberation is perfectly separable from $H_r(\tau)$ even though m and σ are unknown. The process of extracting the unknown $H_r(\tau)$ is completed by finding the inverse logarithm of the inverse Fourier transform of $H_r(\tau)$.

One might wonder whether the above result holds for other pdf's. In general for any pdf, $Q(\omega)$ can be expressed in terms of its (noncentral) moments, α_i , as [3,4].

$$Q(\omega) = 1 + \sum_{i=1}^{\infty} \frac{\alpha_i}{i!} (j\omega)^i \quad (2.4.7)$$

where

$$\alpha_j = E \{ \tau_i^j \} \quad (2.4.8)$$

Using the identity

$$\log(1+z) = \frac{z}{1} - \frac{z^2}{2} + \dots, \quad (2.4.9)$$

letting $z = \sum_{i=1}^{\infty} \frac{\alpha_i}{i!} (j\omega)^i$, and simplifying results in

$$\log Q(\omega) = \sum_{i=1}^{\infty} \frac{\chi_i}{i!} (j\omega)^i \quad (2.4.10)$$

In (2.4.10) the χ 's are the cumulants or semi-invariants of the distribution. If μ_i are the central moments and α_i are the moments, they are related to the cumulants by

$$1 + \sum_{i=1}^{\infty} \frac{\alpha_i}{i!} (j\omega)^i = e^{\sum_{i=1}^{\infty} \frac{\chi_i}{i!} (j\omega)^i} \quad (2.4.11)$$

which by equating like powers breaks down to

$$\begin{aligned}\chi_1 &= \alpha_1 = m \\ \chi_2 &= \alpha_2 - \alpha_1^2 = \sigma^2 \\ \chi_3 &= \alpha_3 - 3\alpha_1\alpha_2 + 2\alpha_1^3 \\ \chi_4 &= \alpha_4 - 3\alpha_2^2 - 4\alpha_1\alpha_3 + 12\alpha_1^2\alpha_2 - 6\alpha_1^4 \\ &\vdots\end{aligned}$$

This can be written conversely as

$$\begin{aligned}\alpha_1 &= \chi_1 \\ \alpha_2 &= \chi_2 + \chi_1^2 \\ \alpha_3 &= \chi_3 + 3\chi_1\chi_2 + \chi_1^3 \\ \alpha_4 &= \chi_4 + 3\chi_2^2 + 4\chi_1\chi_3 + 6\chi_1^2\chi_2 + \chi_1^4 \\ &\vdots\end{aligned}$$

Finally the cumulants χ_i can be expressed in terms of the central moments μ_j where

$$\mu_j = E[(\tau_1 - m)^j] = \int_{-\infty}^{\infty} (\tau - m)^j p_r(\tau) d\tau \quad (2.4.12)$$

as

$$\begin{aligned}\chi_1 &= m \\ \chi_2 &= \sigma^2 \\ \chi_3 &= \mu_3 \\ \chi_4 &= \mu_4 - 3\mu_2^2 \\ \chi_5 &= \mu_5 - 10\mu_2\mu_3 \\ \chi_6 &= \mu_6 - 15\mu_2\mu_4 - 10\mu_3^2 + 30\mu_2^3 \\ &\vdots\end{aligned} \quad (2.4.13)$$

Substituting (2.4.10) into (2.4.1) gives

$$\log \overline{Y(\omega)} = \log H_r(\omega) + \sum_{i=1}^{\infty} \frac{\chi_i}{i!} (j\omega)^i + \log \left[\sum_i \overline{a_i} \right] \quad (2.4.14)$$

Remembering the Fourier transform theorem

$$(j\omega)^n S(\omega) \leftrightarrow \frac{d^n s(\tau)}{d\tau^n} \quad (2.4.15)$$

when taking the transform or cepstrum of (2.4.14) yields

$$Y(\tau) = H_r(\tau) + \sum_{i=1}^{\infty} \frac{\chi_i}{i!} \frac{d^i \delta(\tau)}{d\tau^i} + \log \left[\sum_i \overline{a_i} \right] \delta(\tau) \quad (2.4.16)$$

which again simplifies to

$$Y(\tau) = H_r(\tau) \quad ; \quad \tau > 0 \quad (2.4.17)$$

regardless of the distribution on the impulse time delays.

Reiterating the conclusion we therefore find that even in the most general case of pdf's representable by Graham-Charlier or Edgeworth series [3] the log of the characteristic function is so smooth that its transform inverse produces only singular functions (such as delta, doublet or triplet functions). Under such circumstances perfect separation of reverberation and signal statistics is possible at least in the continuous domain.

2.5 Signal Dereverberation for Source with Dereverberation Between Amplitude and Arrival Time

For the case when the source pulse is not such that the amplitudes, a_i , and the arrival times, τ_i , of the pulses are independent, Equation (2.2.5) cannot be reduced to (2.2.8). Instead, the quantity inside the summation must be evaluated directly.

In general, an expression to be averaged can be related to the pdf of the individual variables, and in particular

$$\overline{a_i e^{-j\omega\tau_i}} = \iint_{-\infty}^{\infty} a_i e^{-j\omega\tau_i} p_r(a_i, \tau_i) d\tau_i da_i \quad (2.5.1)$$

Assuming $p_r(...)$ is Gaussian with $\rho \neq 0$, (2.5.1) can be evaluated by multiplying by $\frac{1}{-j}$ the derivative with respect to ω' of the characteristic equation of a two-dimensional Gaussian function with ω' set to zero. That is,

$$\begin{aligned} & \iint_{-\infty}^{\infty} a e^{-j\omega\tau} p_r(a, \tau) d\tau da \\ &= \frac{1}{-j} \frac{d}{d\omega'} \iint_{-\infty}^{\infty} \frac{1}{2\pi|M|} \exp\left\{ \frac{-[(a - \bar{a})^2 \sigma_{11}^2 - 2\rho a\tau + (\tau - \bar{\tau})^2 \sigma_{22}^2]}{(\sigma_{11}\sigma_{22} - \rho^2)} \right\} \\ & \quad e^{-j\omega\tau} e^{-j\omega' a} d\tau da \Big|_{\omega' = 0} \end{aligned} \quad (2.5.2)$$

Therefore

$$\overline{a e^{-j\omega\tau}} = \frac{1}{-j} \left\{ \frac{d}{d\omega'} \exp [-(\sigma_{11}^2 \omega'^2 + 2\rho\omega\omega' + \sigma_{22}^2 \omega^2)/2 - j(\bar{\tau}\omega + \bar{a}\omega')] \right\} \Big|_{\omega' = 0} \quad (2.5.3)$$

$$= j[-\sigma_{11}^2 (2\omega')/2 - 2\rho\omega/2 - j\bar{a}]$$

$$\exp[-(\sigma_{11}^2 \omega'^2 + 2\rho\omega\omega' + \sigma_{22}^2 \omega^2)/2 - j(\bar{\tau}\omega + \bar{a}\omega')] \Big|_{\omega' = 0} \quad (2.5.4)$$

$$= [\bar{a} - j\rho\omega] e^{-\sigma_{22}^2 \omega^2/2 - j\bar{\tau}\omega} \quad (2.5.5)$$

$$= [\bar{a} - j\rho\omega] Q(\omega) \quad (2.5.6)$$

Substituting back gives

$$\sum_i \overline{a_i} e^{-j\omega\tau_i} = Q(\omega) \sum_i [\bar{a}_i - j\rho\omega] \quad (2.5.7)$$

and

$$H_r(\omega) = \frac{\overline{Y(\omega)}}{Q(\omega) \sum_i [\bar{a}_i - j\rho\omega]} \quad (2.5.8)$$

It should be noted here that (2.5.8) reduces to the same results obtained previously when it was assumed a_i and τ_i are independent, i.e., $\rho = 0$. Also to be noted is that the assumption that $p_r(\dots)$ is Gaussian is not overly restrictive since similar derivations follow for many non-Gaussian probabilities which can be represented by Gram-Charlies or Edgeworth series [3] of normal functions.

Using cepstral analysis, equation (2.4.1) becomes

$$\log \overline{Y(\omega)} = \log H_r(\omega) + \log Q(\omega) + \log \sum_{i=1}^M [\bar{a}_i - j\rho\omega] \quad (2.5.9)$$

$$= \log H_r(\omega) + \log Q(\omega) + \log M\bar{a}_i + \log[1 - j \frac{\rho\omega}{\bar{a}_i}] \quad (2.5.10)$$

and

$$\begin{aligned} y(\tau) = H_r(\tau) + m \frac{d}{d\tau} \delta(\tau) + \frac{\sigma^2}{2} \frac{d^2}{d\tau^2} \delta(\tau) + M\bar{a}_i \delta(\tau) \\ + F^{-1}\{\log[1 - j \frac{\rho\omega}{\bar{a}_i}]\} \end{aligned} \quad (2.5.11)$$

Now the only term to be evaluated is $F^{-1}\{\log[1 - j \frac{\rho\omega}{\bar{a}_i}]\}$.

Using the Taylor series for a complex number y

$$\log(1 - y) = - \sum_{n=1}^{\infty} \frac{y^n}{n} \quad ; \quad |y| < 1 \quad (2.5.12)$$

and recognizing

$$\log(1 - y) = -\log\left(\frac{1}{1-y}\right) \quad (2.5.13)$$

then

$$\log \left(\frac{1}{1-y} \right) = \sum_{n=1}^{\infty} \frac{y^n}{n} \quad (2.5.14)$$

Employing the transformation for a complex z ,

$$z = \frac{1}{1-y} \quad (2.5.15)$$

implies

$$\frac{z-1}{z} = y \quad (2.5.16)$$

and

$$\log(z) = \sum_{n=1}^{\infty} \frac{\left(\frac{z-1}{z} \right)^n}{n} ; \left| \frac{z-1}{z} \right| < 1 \quad (2.5.17)$$

If $z = 1 + jx$, the condition $\left| \frac{z-1}{z} \right| < 1$ requires

$$\sqrt{\frac{x^2 + x^2}{1 + 2x^2 + x^4}} < 1 \quad (2.5.18)$$

or

$$2x^2 < 1 + 2x^2 + x^2 \quad (2.5.19)$$

or else

$$1 < x^4 \quad (2.5.20)$$

which is true for all real x and so (2.5.17) is a valid series for $\log(z)$.

Using the expansion of (2.5.17) on the term $\log [1 - j \frac{\rho\omega}{\bar{a}_i}]$

yields

$$\log [1 - j \frac{\rho\omega}{\bar{a}_i}] = \sum_{k=1}^{\infty} \frac{1}{k} (j\omega)^k \left(\frac{-\rho / \bar{a}_i}{1 - j \frac{\rho\omega}{\bar{a}_i}} \right)^k \quad (2.5.21)$$

$$= (j\omega) \left(\frac{-\rho / \bar{a}_i}{1 - j \frac{\rho\omega}{\bar{a}_i}} \right) + \frac{(j\omega)^2}{2} \left(\frac{-\rho / \bar{a}_i}{1 - j \frac{\rho\omega}{\bar{a}_i}} \right)^2 + \dots \quad (2.5.22)$$

and taking the inverse transform

$$F^{-1}\{\log[1 - j \frac{\rho\omega}{\bar{a}_i}]\} = \frac{d}{d\tau} F^{-1}\left\{\left[\frac{-\rho / \bar{a}_i}{1 - j \frac{\rho\omega}{\bar{a}_i}}\right]\right\} + \frac{1}{2} \frac{d^2}{d\tau^2} F^{-1}\left\{\left[\frac{-\rho / \bar{a}_i}{1 - j \frac{\rho\omega}{\bar{a}_i}}\right]^2\right\} + \dots \quad (2.5.23)$$

$$= \frac{d}{d\tau} f(\tau) + \frac{1}{2} \frac{d}{d\tau} [f(\tau) * f(\tau)] + \dots \quad (2.5.24)$$

where

$$f(\tau) = \exp(\bar{a}_i \tau / \rho) U(\tau) \quad (2.5.25)$$

Equation (2.5.24) can be rewritten

$$F^{-1}\{\log[1 - j \frac{\rho\omega}{\bar{a}_i}]\} = g(\tau) + \frac{1}{2} g(\tau) * g(\tau) + \dots \quad (2.5.26)$$

with

$$g(\tau) = \frac{d}{d\tau} f(\tau) = \exp(\bar{a}_i \tau / \rho) U(\tau) + \frac{\bar{a}_i}{\rho} \delta(\tau) \quad (2.5.27)$$

Now $g(\tau)$ will not blow up if \bar{a}_i and ρ , the correlation coefficient between the amplitude and arrival time statistics, have opposite signs. $F^{-1}\{\log[1 - j \frac{\rho\omega}{a_1}]\}$ will have significant value as τ decreases to zero and will destroy more information about $H_r(\tau)$ for small τ .

2.6 Signal Dereverberation by Use of Ad Hoc Two-Dimensional Probability Distribution Functions

If the one-dimensional pdf approach taken in Section 2.3 is not appropriate (i.e., $\overline{Y(\omega)} = 0$, $\sum_i \bar{a}_i = 0$), the problem can be restated in terms of its two-dimensional statistics. Equation (2.2.3) becomes

$$Y(\omega)Y(\omega') = H_r(\omega)H_r(\omega') \left[\sum_i a_i e^{-j\omega\tau_i} \right] \left[\sum_i a_i e^{-j\omega'\tau_i} \right] \quad (2.6.1)$$

Using the same technique as in Section 2.2, (2.2.4), (2.2.6) and (2.2.8) become

$$Y(\omega)Y(\omega') = H_r(\omega)H_r(\omega') \overline{\left[\sum_i a_i e^{-j\omega\tau_i} \right] \left[\sum_i a_i e^{-j\omega'\tau_i} \right]} \quad (2.6.2)$$

$$= H_r(\omega)H_r(\omega') \left[\sum_i \sum_k \overline{a_i a_k} e^{-j(\omega\tau_i + \omega'\tau_k)} \right] \quad (2.6.3)$$

$$= H_r(\omega)H_r(\omega') \hat{Q}(\omega, \omega') \left[\sum_i \sum_k \overline{a_i a_k} \right] \quad (2.6.4)$$

The last equation can be rewritten if ω and ω' are independent in \hat{Q} as

$$\overline{Y(\omega)Y(\omega')} = H_r(\omega)H_r(\omega')Q(\omega)Q'(\omega') \left[\sum_i \sum_k \overline{a_i a_k} \right] \quad (2.6.5)$$

for which the one-dimensional statistics could still be used.

Finding a suitable $\hat{Q}(\omega, \omega')$ can be done again by using some ad hoc pdf functions. A few of these are given in Table 2.6.1 along with their characteristic functions. Further simplifications occur when $\omega = \omega'$. These are given in Table 2.6.2.

1) Bivariate, Jointly Normal

$$Q(\omega)Q(\omega') = e^{-(\omega^2 + \omega'^2 + 2\omega\omega'\rho)\sigma^2/2}$$

$$q(t, t') = \frac{1}{2\pi\sigma\sigma'\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)} \left[\frac{(t-\bar{t})^2}{\sigma^2} - \frac{2\mu(t-\bar{t})(t'-\bar{t}')}{\sigma\sigma'} + \frac{(t'-\bar{t}')^2}{\sigma'^2} \right]}$$

$$\rho = \frac{\mu}{\sigma\sigma'}, \quad \mu = \text{covariance}$$

$$[q(t, t') = \frac{\pm m \sqrt{1-\rho^2}}{\pi k^m} (k \mp t^2 + 2\rho t t' \mp t'^2)^{m-1}]$$

$$m > 0, \quad |\rho| < 1, \quad k > 0$$

2) Laplace

$$Q(\omega)Q'(\omega') = \frac{1}{1 + (\omega^2 + \omega'^2 + 2\omega\omega'\rho)\frac{\sigma^2}{2}}$$

$$q(t, t') = \frac{1}{\pi\sigma^2\sqrt{1-\rho^2}} K_0(\sqrt{2}l/\sigma)$$

3) Quasi-Pyramid

$$Q(\omega)Q'(\omega') = \frac{\alpha}{cb} \operatorname{sinc}^2\left(\frac{\omega}{2\pi c}\right) \operatorname{sinc}^2\left(\frac{\omega'}{2\pi b}\right)$$

$$q(t, t') = \alpha \Lambda\left(\frac{t}{c}\right) \Lambda\left(\frac{t'}{b}\right)$$

TABLE 2.6.1 Ad Hoc Functions for $Q(\omega)Q'(\omega')$ [3,4,5]

4) Pearson Type VII

$$Q(\omega)Q'(\omega') = \frac{2^{1-\nu}}{\Gamma(\nu)} (a\sqrt{\omega^2 + \omega'^2 + 2\omega\omega'\rho})^\nu K_\nu(a\sqrt{\omega^2 + \omega'^2 + 2\omega\omega'\rho})$$

$$\nu > 0, \quad a > 0$$

$$q(t, t') = \frac{\nu}{\pi a^{-2\nu} \sqrt{1-\rho^2}} (a^2 + \ell^2)^{-\nu-1}$$

5) Uniform

$$Q(\omega)Q'(\omega') = \frac{1}{cb} \frac{\sin(c\omega) \cdot \sin(b\omega')}{\omega\omega'}$$

$$q(t, t') = \frac{1}{4cb}$$

$$|t| < c, \quad |t'| < b$$

6) Pearson Type II

$$Q(\omega)Q'(\omega') = 2^\nu \Gamma(\nu+1) \frac{J_\nu(A\sqrt{\omega^2 + \omega'^2 + 2\omega\omega'\rho})}{(A\sqrt{\omega^2 + \omega'^2 + 2\omega\omega'\rho})^\nu}$$

$$\nu > 0$$

$$q(t, t') = \frac{\nu}{\pi A^{2\nu} \sqrt{1-\rho^2}} (A^2 + \ell^2)^{\nu-1}$$

$$|\ell| < A, \quad \nu > 0, \quad \ell = \frac{\sqrt{t^2 + 2tt' + t'^2}}{1 - \rho^2}$$

$J_\nu(\cdot)$ = Bessel function of 1st kind

$K_\nu(\cdot)$ = Modified Bessel function of 2nd kind

TABLE 2.6.1 Ad Hoc Functions for $Q(\omega)Q'(\omega')$ (continued)

$$1) \quad Q(\omega)Q'(\omega) = e^{-(2 + 2\rho)\omega^2\sigma^2/2}$$

NORMAL

$$2) \quad Q(\omega)Q'(\omega) = \frac{1}{1 + (1 + \rho)\omega^2\sigma^2}$$

LAPLACE

$$3) \quad Q(\omega)Q'(\omega) = 2^\nu \Gamma(\nu + 1) \frac{J_\nu(A\omega\sqrt{2 + 2\rho})}{(A\omega\sqrt{2 + 2\rho})}$$

PEARSON TYPE II

$$\nu > 0$$

$$4) \quad Q(\omega)Q'(\omega) = \frac{2^{1-\nu}}{\Gamma(\nu)} (a\omega\sqrt{2 + 2\rho})^\nu K_\nu(a\omega\sqrt{2 + 2\rho})$$

PEARSON TYPE VII

$$\nu > 0$$

$$a > 0$$

$$5) \quad Q(\omega)Q'(\omega) = \frac{1}{ab} \frac{\sin a\omega \sin b\omega}{\omega^2}$$

UNIFORM

$$= \frac{1}{a^2} \frac{\sin^2 a\omega}{\omega^2} \quad \text{if } a = b$$

TABLE 2.6.2 Ad Hoc Functions for $Q(\omega)Q(\omega')$ When $\omega = \omega'$

2.7 Signal Dereverberation through Homomorphic Filtering with Two-Dimensional Statistics

Applying homomorphic filtering to the two-dimensional statistics first requires taking the log of both sides of (2.6.5) to get

$$\log \{\overline{Y(\omega)Y(\omega')}\} = \log H_r(\omega) + \log H_r(\omega') + \log Q(\omega) + \log Q'(\omega') + \log \left[\sum_i \sum_k \overline{a_i a_k} \right] \quad (2.7.1)$$

assuming τ_i independent of τ_k , a_i and a_k . Substituting (2.4.10) into (2.7.1) will give

$$\log \{\overline{Y(\omega)Y(\omega')}\} = \log H_r(\omega) + \log H_r(\omega') + \sum_{i=1}^{\infty} \frac{\chi_i}{i!} (j\omega)^i + \sum_{i=1}^{\infty} \frac{\chi'_i}{i!} (j\omega')^i + \log \left[\sum_i \sum_k \overline{a_i a_k} \right]$$

For the special case $\omega = \omega'$ (2.7.2)

$$\log \{\overline{Y^2(\omega)}\} = 2 \log H_r(\omega) + \sum_{i=1}^{\infty} \frac{\chi_i + \chi'_i}{i} (j\omega)^i + \log \left[\sum_i \sum_k \overline{a_i a_k} \right] \quad (2.7.3)$$

whose cepstrum again for $\tau \neq 0$ simplifies to

$$V(\tau) = 2H_r(\tau) \quad \tau > 0 \quad (2.7.4)$$

where now $V(\tau)$ is the inverse transform of $\log \{\overline{Y^2(\omega)}\}$. In general, going back to (2.6.4), similar analysis can be done to get

$$\begin{aligned} \log \{\overline{Y(\omega)Y(\omega')}\} &= \log H_r(\omega) + \log H_r(\omega') + \sum_{i=1}^{\infty} \sum_{k=1}^{\infty} \frac{\chi_{ik}}{i!k!} (j\omega)^i (j\omega')^k \\ &+ \log \left[\sum_i \sum_k \overline{a_i a_k} \right] \end{aligned} \quad (2.7.5)$$

(For the case of nonindependence between τ_i and τ_k , but independence with a_i , a_k) given that $\log Q(\omega, \omega')$ can be expressed from Section 2.4

$$\log Q(\omega, \omega') = \sum_{i=1}^{\infty} \sum_{k=1}^{\infty} \frac{\chi_{ik}}{i!k!} (j\omega)^i (j\omega')^k \quad (2.7.6)$$

This can be expanded and the cepstrum taken to give similar results to (2.7.4) for $\tau \neq 0$.

In this case the equation corresponding to (2.6.5) becomes

$$\overline{Y(\omega)Y(\omega')} = H_r(\omega)H_r(\omega') Q(\omega, \omega') \left[\sum_i \sum_k \overline{a_i a_k} \right] \quad (2.7.7)$$

So that if $Q(\omega, \omega')$ is known a-priori

$$H_r(\omega)H_r(\omega') = \frac{\overline{Y(\omega)Y(\omega')}}{Q(\omega, \omega') \left[\sum_i \sum_k \overline{a_i a_k} \right]} \quad (2.7.8)$$

resulting in

$$H_r^2(\omega) = \frac{\overline{Y^2(\omega)}}{Q(\omega, +\omega) \left[\sum_i \sum_k \overline{a_i a_k} \right]} \quad (2.7.9)$$

for $\omega' = +\omega$. The resulting cepstrum is once again

$$Y(\tau) = 2H_r(\tau) \quad ; \quad \tau \neq 0 \quad (2.7.10)$$

It is noted here that for both the one and two dimensional statistics, once $H_r(\omega)$ or $H_r(\omega)H_r(\omega')$ has been found, the received signal is simply multiplied by the inverse of H_r to dereverberate it. For the multi-element array, $H_{ri}(\omega)$, $i = 1, 2, \dots, N$ must be found for each of the N elements and the signal received by the i^{th} element must be multiplied by $H_{ri}^{-1}(\omega)$.

Each $H_r(\omega)$ is found from the previously computed $H_r(\tau)$ which is obtained from the measured quantity $Y(\tau)$ by the relation

$$H_r(\omega) = F\{\hat{H}_r(\tau)\} \quad (2.7.11)$$

where

$$\hat{H}_r(\tau) = \begin{cases} 0 & ; -\epsilon < \tau < \epsilon \\ \hat{H}_r(\tau) & ; \text{elsewhere} \end{cases} \quad (2.7.12)$$

$\hat{H}_r(\tau)$ is an estimate of $H_r(\tau)$ obtained by either zeroing $H_r(\tau)$ for small τ or extrapolating $H_r(\tau)$ for $-\epsilon < \tau < \epsilon$ from $H_r(\tau)$ outside the region of divergence.

2.8 Treatment of Hermite Expansion Source Pulse Case

In this section the assumption that the signal pulse is an ideal impulse will be removed and generalized. Rewriting eq. (2.2.3) in terms of a general pulse shape $h_p(t)$ or $H_p(\omega)$ gives

$$Y(\omega) = H_r(\omega) X(\omega) \quad (2.8.1)$$

$$= H_r(\omega) \left[\sum_i H_{pi}(\omega) e^{-j\omega\tau_i} \right] \quad (2.8.2)$$

Averaging both sides as before and assuming independence between arrival time and pulse shape amplitude

$$\overline{Y(\omega)} = H_r(\omega) \overline{\left[\sum_i H_{pi}(\omega) e^{-j\omega\tau_i} \right]} \quad (2.8.3)$$

$$= H_r(\omega) Q(\omega) \sum_i \overline{H_{pi}(\omega)} \quad (2.8.4)$$

where

$$\overline{H_{pi}(\omega)} = \int_{-\infty}^{\infty} \overline{h_{pi}(t)} e^{-j\omega t} dt \quad (2.8.5)$$

is the Fourier transform of the average pulse shape. Here $\overline{h_{pi}(t)}$ will not be taken to be an ideal delta function however it will be assumed to have a Hermite orthogonal expansion [4], i.e.,

$$\overline{h_{pi}(t)} = \sum_{\ell=0}^{\infty} \left[-\frac{\kappa_1^{\ell}}{1! \ell!} D^{\ell} + \frac{\kappa_2^{\ell}}{2! \ell!} D^{2\ell} - \frac{\kappa_3^{\ell}}{3! \ell!} D^{3\ell} + \frac{\kappa_4^{\ell}}{4! \ell!} D^{4\ell} \dots \right] \alpha(t) \quad (2.8.6)$$

$$= \sum_{\ell=0}^{\infty} \left[\sum_{m=1}^{\infty} (-1)^m \frac{\kappa_m^{\ell}}{m! \ell!} D^{m\ell} \right] \alpha(t) \quad (2.8.7)$$

where

$$D = \frac{d}{dt}, \quad (2.8.8)$$

$$\alpha(t) = \frac{e^{-t^2/2}}{\sqrt{2\pi}} \quad (2.8.9)$$

and the κ_i are the "cumulants" of the pulse shape. Eq. (2.8.7) can be shown to represent a sum of orthogonal terms since

$$(-D)^{\ell} \alpha(t) = H_{\ell}(t) \alpha(t) \quad (2.8.10)$$

where $H_{\ell}(t)$ is the ℓ^{th} Tchebycheff-Hermite polynomial.

Given $\overline{h_{pi}(t)}$, its Fourier transform is

$$\overline{H_{pi}(\omega)} = e^{-\omega^2/2} \exp\left[\frac{\kappa_1}{1!} j\omega + \frac{\kappa_2}{2!} (j\omega)^2 + \frac{\kappa_3}{3!} (j\omega)^3 + \frac{\kappa_4}{4!} (j\omega)^4 \dots\right] \quad (2.8.11)$$

$$= e^{-\omega^2/2} \exp\left[\sum_{m=1}^{\infty} \frac{\kappa_m}{m!} (j\omega)^m\right] \quad (2.8.12)$$

Calculating the log of $\overline{H_{pi}(\omega)}$ is straightforward and its cepstrum is

$$H_p(\tau) = \frac{1}{2} \frac{d^2}{d\tau^2} \delta(\tau) + \sum_{m=1}^{\infty} \frac{\kappa_m}{m!} \frac{d^m}{d\tau^m} \delta(\tau) \quad (2.8.13)$$

This again shows the cepstrum of the noise pulse is confined to the region $\tau = 0$ in the cepstrum or $|\tau| < \epsilon$ in the digital approximations to the continuous domain. This shows there are a broad range of pulse shapes which fit the orthogonal expansion and still are confined to the origin in the cepstrum of the received signal.

Applying this same assumption and analysis to the two dimensional probability case, which must be used when the average pulse shape is zero, gives similar results. Eq. (2.6.1) can be generalized to

$$Y(\omega) Y(\omega') = H_r(\omega) H_r(\omega') \left[\sum_i H_{pi}(\omega) e^{-j\omega\tau_i} \right] \left[\sum_k H_{pk}(\omega') e^{-j\omega'\tau_k} \right] \quad (2.8.14)$$

With the previous assumption and analysis one proceeds as follows;

$$\overline{Y(\omega) Y(\omega')} = H_r(\omega) H_r(\omega') \overline{\left[\sum_i H_{pi}(\omega) e^{-j\omega\tau_i} \right] \left[\sum_k H_{pk}(\omega') e^{-j\omega'\tau_k} \right]} \quad (2.8.15)$$

$$= H_r(\omega) H_r(\omega') \overline{\left[\sum_i \sum_k H_{pi}(\omega) H_{pk}(\omega') e^{-j(\omega\tau_i + \omega'\tau_k)} \right]} \quad (2.8.16)$$

$$= H_r(\omega) H_r(\omega') \hat{Q}(\omega, \omega') \overline{\left[\sum_i \sum_k H_{pi}(\omega) H_{pk}(\omega') \right]} \quad (2.8.17)$$

$$= H_r(\omega) H_r(\omega') Q(\omega) Q'(\omega') \overline{\left[\sum_i \sum_k H_{pi}(\omega) H_{pk}(\omega') \right]} \quad (2.8.18)$$

where

$$\overline{H_{pi}(\omega) H_{pk}(\omega')} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \overline{h_{pi}(t) h_{pk}(t')} e^{-j(\omega t + \omega' t')} dt dt' \quad (2.8.19)$$

and $\overline{h_{pi}(t) h_{pk}(t')}$ is the two dimensional average pulse shape.

This can be represented by a two dimensional Hermite orthogonal expansion [4] of

$$\overline{h_{pi}(t) h_{pk}(t')} = \sum_{\ell=0}^{\infty} \left\{ \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} [(-1)^{m+n} \frac{\kappa_{mn}^{\ell} D_1^{m\ell} D_2^{n\ell}}{m! n! \ell!}] \right\} \frac{1}{2\pi\sqrt{1-\rho^2}} \cdot \exp\left[\frac{-(t^2 - 2\rho t t' + t'^2)}{2(1-\rho^2)}\right] \quad (2.8.20)$$

where

$$D_1 = \frac{d}{dt} \quad , \quad (2.8.21)$$

$$D_2 = \frac{d}{dt'} \quad , \quad (2.8.22)$$

and the κ_{mn} are the two dimensional "cumulants" of the two dimensional statistical average pulse shape. Eq. (2.8.20) can also be shown to be a sum of orthogonal terms of two dimensional Tchebycheff-Hermite polynomials [9].

The transform of eq. (2.8.20) is

$$\overline{H_{pi}(\omega) H_{pk}(\omega')} = \exp\left[-\frac{1}{2} (\omega^2 + 2\rho\omega\omega' + \omega'^2)\right] \exp\left[\sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \kappa_{mn} \frac{(j\omega)^m (j\omega')^n}{m! n!}\right] \quad (2.8.23)$$

By setting $\omega' = \omega$ or $\omega' = -\omega$ the one dimensional cepstrum can be obtained. The first substitution yields

$$H_p(\tau) = -(1 + \rho) \frac{d^2}{d\tau^2} \delta(\tau) + \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{\kappa_{mn}}{m! n!} \frac{d^{m+n}}{d\tau^{m+n}} \delta(\tau) \quad (2.8.24)$$

whereas $\omega' = -\omega$ gives

$$H_p(\tau) = -(1 - \rho) \frac{d^2}{d\tau^2} \delta(\tau) + \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} (-1)^n \frac{\kappa_{mn}}{m! n!} \frac{d^{m+n}}{d\tau^{m+n}} \delta(\tau) \quad (2.8.25)$$

If ρ in eq. (2.8.23) is equal to zero then the two dimensional statistics are separable and eq. (2.8.24) can be rewritten as

$$H_p(\tau) = \frac{d^2}{d\tau^2} \delta(\tau) + 2 \sum_{m=1}^{\infty} \frac{\kappa'_m}{m!} \frac{d^m}{d\tau^m} \delta(\tau) \quad (2.8.26)$$

Again it has been shown that there are a class of pulse shapes into which the source noise pulse can fall and yet the cepstrum of which still is confined to $\tau = 0$. This meets the requirements for perfect separability in using the cepstrum outside the $\tau = 0$ region to calculate the channel response and dereverberate the signal with an inverse channel filter. Hermite expansions exist for all square-integrable functions where $\int_0^{\infty} f^2(t)dt$ exists but this is just a subset of all functions. Many non-Hermite pulse shapes exist which violate this condition.

2.9 Errors Due to General Source Pulse Transfer Function

In this section the errors introduced by modeling the noise as a series of ideal impulse functions rather than finite width pulses will be studied. If $h_p(t)$ represents the actual non-ideal noise pulse, then the source will be

$$v(t) = \sum_i a_i h_{pi}(t - \tau_i) = \sum_i h_{pi}(t) * a_i \delta(t - \tau_i) \quad (2.9.1)$$

Given that

$$H_{pi}(\omega) = \int_{-\infty}^{\infty} h_{pi}(t) e^{-j\omega t} dt \quad (2.9.2)$$

and

$$H_r(\omega) = \int_{-\infty}^{\infty} h_r(t) e^{-j\omega t} dt \quad (2.9.3)$$

where $h_r(t)$ is the channel impulse response, then

$$y(t) = h_r(t) * v(t) = h_r(t) * \sum_i h_{pi}(t) * a_i \delta(t - \tau_i) \quad (2.9.4)$$

and

$$Y(\omega) = H_r(\omega) V(\omega) = H_r(\omega) \left[\sum_i H_{pi}(\omega) a_i e^{-j\omega\tau_i} \right] \quad (2.9.5)$$

Averaging both sides gives

$$\overline{Y(\omega)} = H_r(\omega) \overline{V(\omega)} = H_r(\omega) \left[\sum_i \overline{H_{pi}(\omega) a_i e^{-j\omega\tau_i}} \right] \quad (2.9.6)$$

$$\overline{Y(\omega)} = H_r(\omega) Q(\omega) \sum_i \overline{a_i H_{pi}(\omega)} \quad (2.9.7)$$

assuming $a_i h_{pi}(t)$ and τ_i are independent. Logging both sides and taking the cepstrum yields

$$Y(\tau) = H_r(\tau) + \frac{d}{d\tau} \delta(\tau) + \frac{\sigma^2}{2} \frac{d^2}{d\tau^2} \delta(\tau) + \bar{H}_p(\tau) \quad (2.9.8)$$

Here $\bar{H}_p(\tau)$ represents the cepstrum of the average pulse shape. If

$h_p(t)$ is modeled by a Gaussian pulse

$$h_p(t) = e^{-t^2/2\sigma^2} \quad (2.9.9)$$

there will be no contribution to the cepstrum for $\tau \neq 0$ and for any very narrow pulse, computer simulations have shown that their cepstra are limited to small values of τ . It should be noted here that many seismic surface pulses can be modeled by Lamb functions, a form of logarithmic pulse. This pulse cannot be perfectly reproduced by a Hermite expansion. Cepstral processing on such signals will result in the loss of some information and distortion of the original pulse shape.

2.10 Signal Dereverberation Within A Dispersive Channel

Although cepstral processing has been shown to be very useful for signal dereverberation it does have a few theoretical limitations. This section demonstrates the cases where cepstral techniques break down and fail to give a viable solution for reverberant channels.

To calculate a channel's transfer function it was necessary to throw out the part of the cepstrum around $\tau = 0$ since this contained mostly signal pulse information and little channel information. In general, by starting with an impulse function or any of its derivatives at $\tau = 0$ in the cepstrum and working the problem backwards an expression can be obtained for all the time functions that end up at $\tau = 0$. That is

$$\int_{-\infty}^{\infty} \frac{d^n \delta(\tau)}{d\tau^n} e^{-j\omega\tau} d\tau = (-j\omega)^n \quad (2.10.1)$$

and taking the antilog gives $e^{(-j\omega)^n}$. The inverse transform of this function is difficult to find for n in general but can be evaluated for a specific n . The first several values of the time function, $f(t)$, are:

$$n = 0 \quad ; \quad f(t) = \delta(t) \quad (2.10.2)$$

$$n = 1 \quad ; \quad f(t) = \delta(t-1) \quad (2.10.3)$$

$$n = 2 \quad ; \quad f(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/4} \quad (2.10.4)$$

$$n = 3 \quad ; \quad f(t) = \frac{1}{\sqrt{2\pi}} e^{-(t-1)^2/4} \quad (2.10.5)$$

$$\begin{aligned} n = 4 \quad ; \quad f(t) &= \frac{1}{\sqrt{4\pi}} e^{-t^2/4} * e^{-t^2/4} \quad (2.10.6) \\ &= c e^{-t^2/d} \end{aligned}$$

The values of $f(t)$ for $n = 2$ and $n = 4$ are the critical ones showing that homomorphic filtering will be unable to separate a dispersive channel response from the pulse response. In fact for all $n > 1$ $f(t)$ will simplify down to some form of Gaussian curve. In other words a channel containing a continuum of paths in a given range from source to sensor will not be able to be separated out using homomorphic processing. This case could arise in a nonhomogeneous media whose density varies from point to point and therefore varies the wave propagation velocity.

However there is still a wide range of applications on which cepstral filtering does work. Also there is some ongoing research in the area of Volterra filtering which may have some promising solutions to this problem when homomorphic filtering fails.

2.11 Numerical Considerations of the IDFT of $(j\omega)^i/i!$

Some attention should be paid to the numerical results obtained when calculating the inverse discrete Fourier transform (IDFT) of $\sum_{i=1}^{\infty} (j\omega)^i \frac{x_i}{i!}$. The results obtained in the discrete numerical case will only approximate those obtained in the ideal continuous case. Although as the number of samples increases, the discrete Fourier transform (DFT) case will more closely approximate the continuous case, it is important to be able to analyze and put limits on the maximum error expected.

Numerically $x_i(n)$, the inverse DFT of $(j\omega)^i/i!$, is expressed as

$$x_i(n) = \frac{1}{N} \sum_{k=0}^{N-1} (j2\pi k)^i e^{j\frac{2\pi}{N} kn} \quad 2.11.1$$

$$x_i(n) = \frac{1}{N} \cdot (j2\pi)^i \sum_{k=0}^{N-1} k^i e^{j\frac{2\pi}{N} kn} \quad 2.11.2$$

This cannot in general be expressed analytically so computer simulations of $x_i(n)$ were made for several values of i . Figures 2.11.1 and 2.11.2 show the magnitude and phase of the IDFT of $j\omega$ using a 256 point transform. The results are poor due to the small number of points and there is a significant contribution for all values of time when compared to the ideal continuous case, an impulse function, which equals zero for all value of time $t \neq 0$. The phase plot shows the rapid variation in the phase of this non-real time function. The ideal case of the continuous inversion would be purely real and exist only at $t = 0$. Figure 2.11.2 would be zero throughout in this case. The magnitude response can be improved by increasing the number of points, however, the phase will retain the same form. This will not be a problem since the phase will be due to very small but non-zero real and imaginary terms.

The plots in this section and also section 2.12 have been designed to be used on fast Fourier transform (FFT) routines. Accordingly they exhibit the wraparound phenomenon of these types of plots where the upper half of the x-axis is merely a translation of the negative portion of the plot to a positive abscissa position.

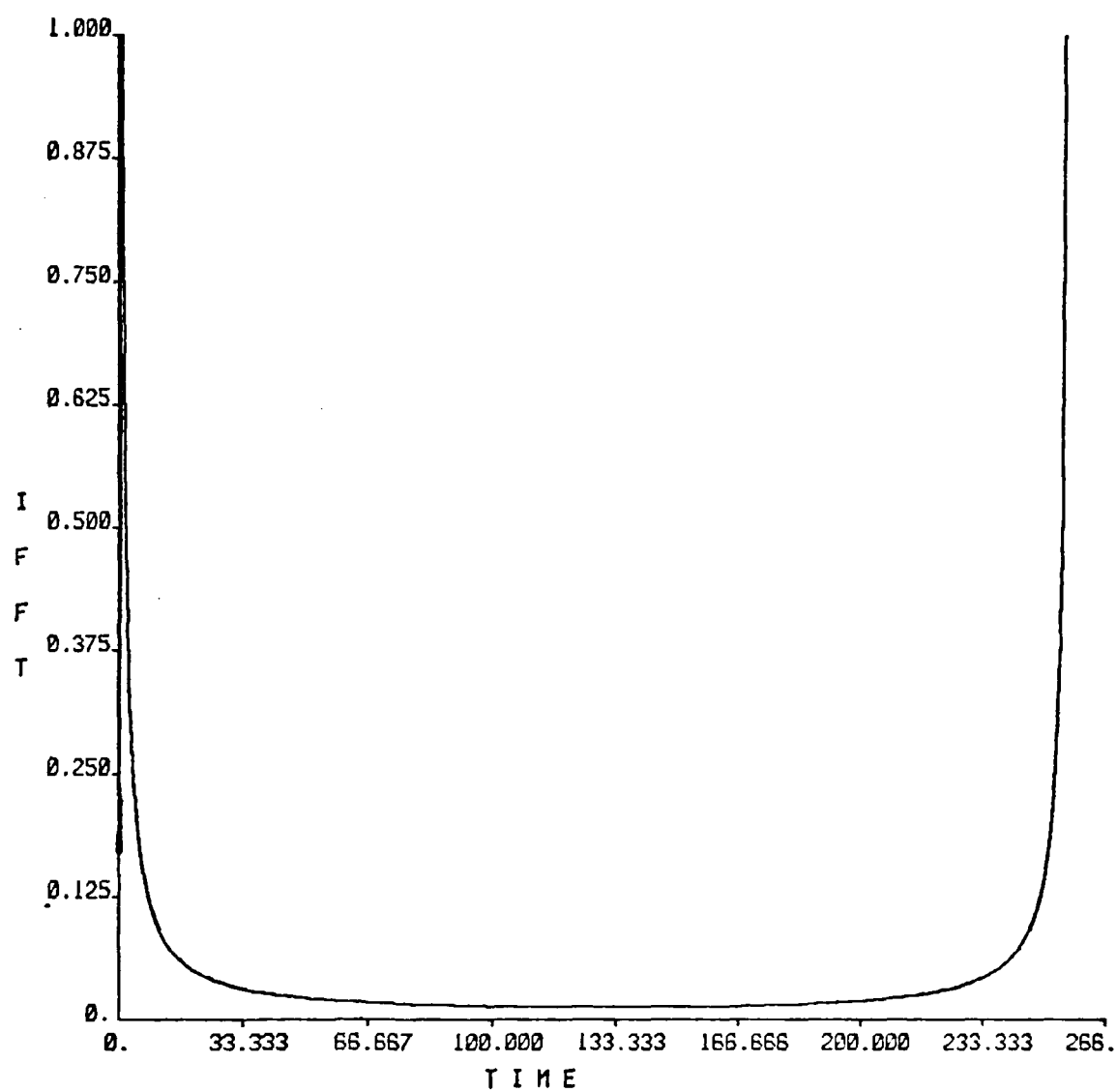


Figure 2.11.1 Magnitude of Inverse Fourier Transform of $j\omega$

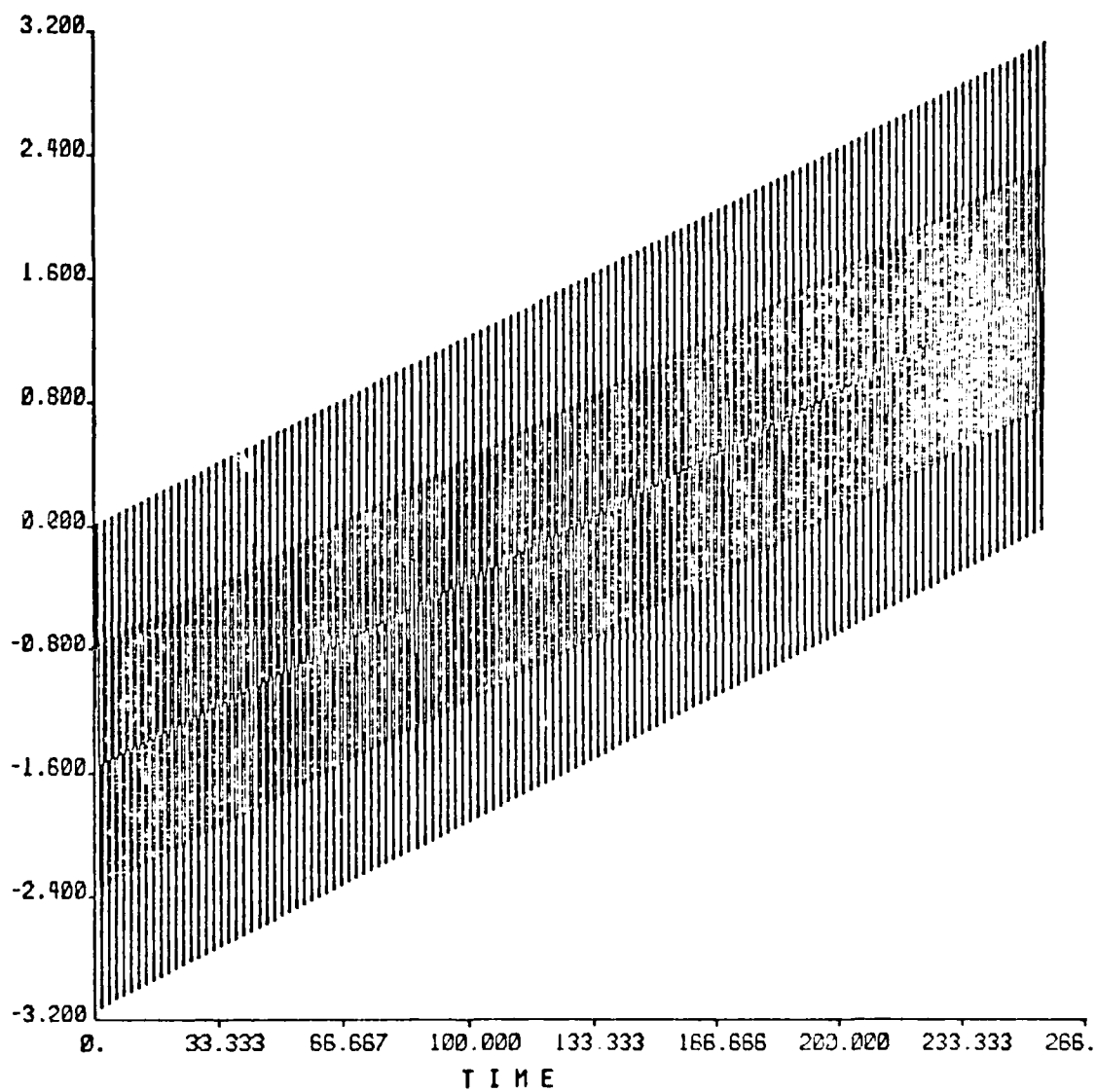


Figure 2.11.2 Phase of Inverse Fourier Transform of $j\omega$

Increasing the number of points to 2048, Figures 2.11.3 and 2.11.4 show the new results for the same function. This shows the improvement in the magnitude response due to the increased sample frequency. Here there is no significant contribution for a normalized radian frequency greater than $\pi/4$ and further improvement can be obtained by increasing the number of points. The phase is as expected, still varying rapidly.

Figures 2.11.5 and 2.11.6 show the magnitude of the IFFT of $(j\omega)^5/5!$ and $(j\omega)^9/9!$ respectively with their absolute scales given. Finally Figures 2.11.7 and 2.11.8 show all three magnitude plots on the same axis. The functions are $j\omega$, $(j\omega)^5/5!$, and $(j\omega)^9/9!$ with Figure 2.11.7 being unlogged and normalized and Figure 2.11.8 just being logged to show the details. Obviously from Figures 2.11.7 and 2.11.8, the most significant contributions will be due to $j\omega$. In 2.11.7, the other two functions are barely resolvable due to their relatively small magnitudes when compared to the IDFT of $j\omega$.

It can be seen that the factorial term rapidly decreases the contributions of $(j\omega)^i/i!$ as i increases. The cumulants, χ_i , which scale the $(j\omega)^i/i!$ term also go to zero for increasing i with the speed of convergence depending on the distribution function being analyzed. In general, the series $\sum_{i=1}^{\infty} (j\omega)^i \frac{\chi_i}{i!}$ should converge fairly rapidly. To be noted here is that this major contribution of error is due to the $j\omega$ term and if this error is minimized by using a large enough number of points for the DFT, the rest of the series' contributions will be negligible.

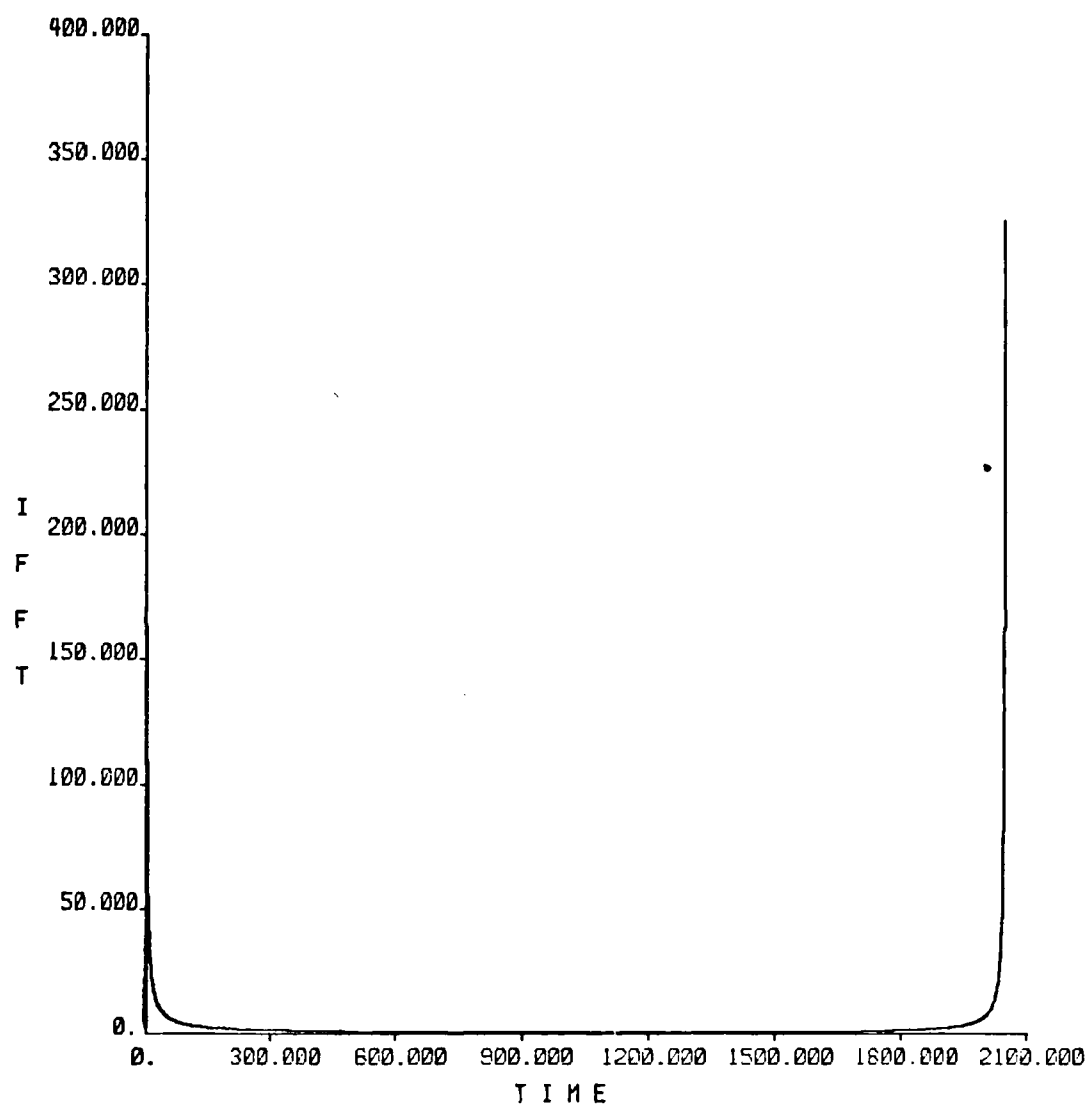


Figure 2.11.3 Magnitude of Inverse Fourier Transform of $j\omega$

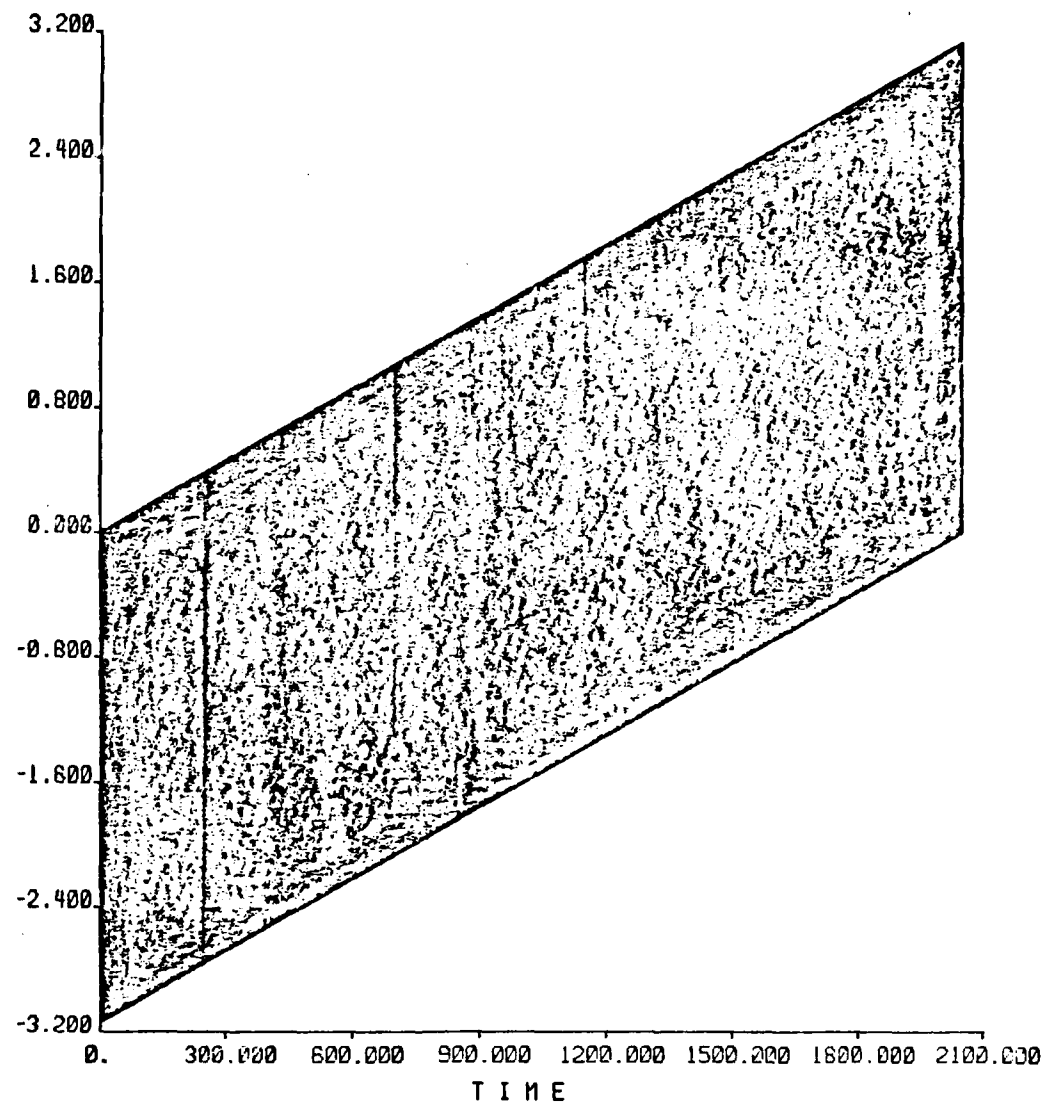


Figure 2.11.4 Phase of Inverse Fourier Transform of $j\omega$

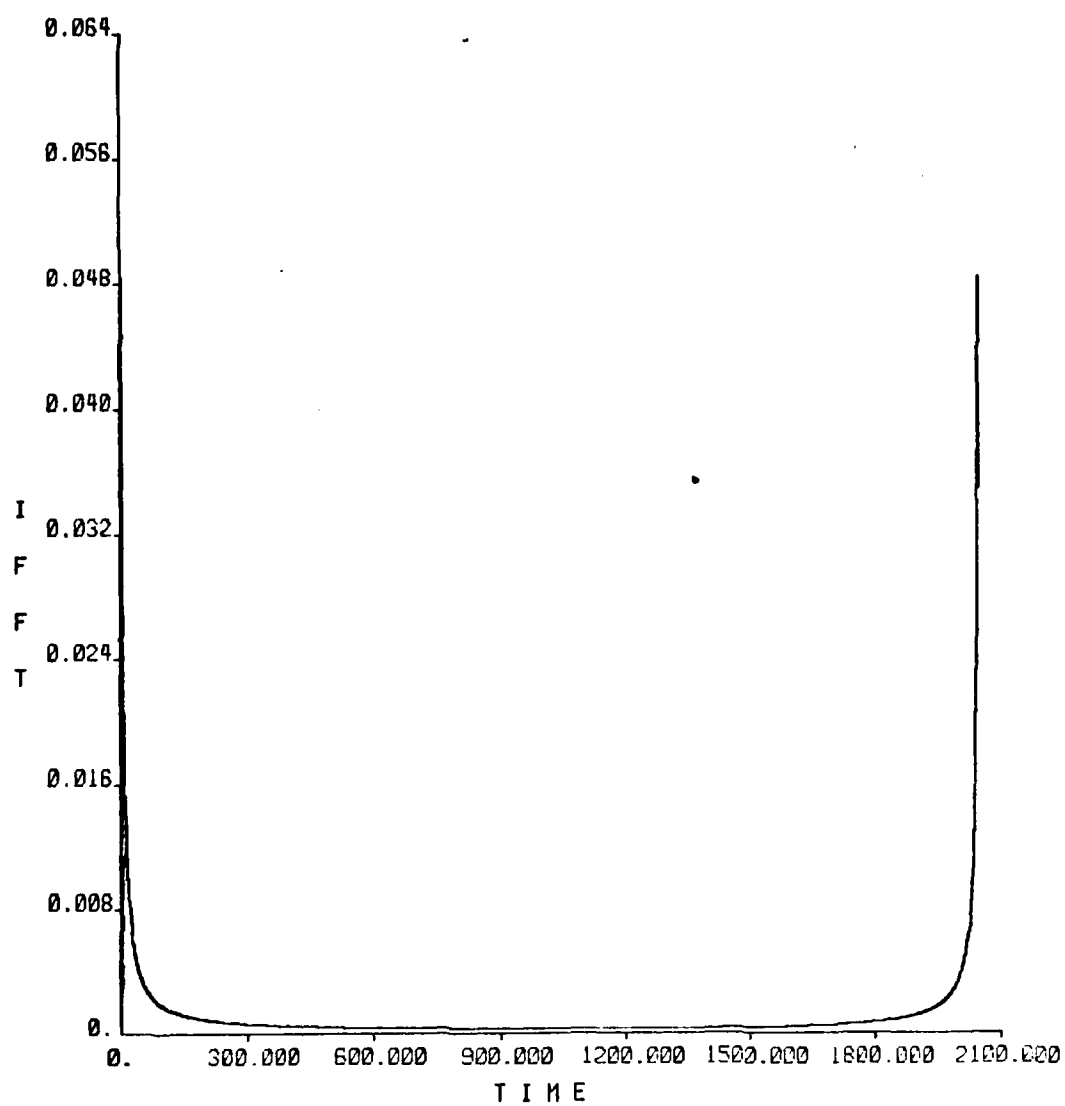


Figure 2.11.5 Magnitude of Inverse Fourier Transform of $\frac{(j\omega)^5}{5!}$

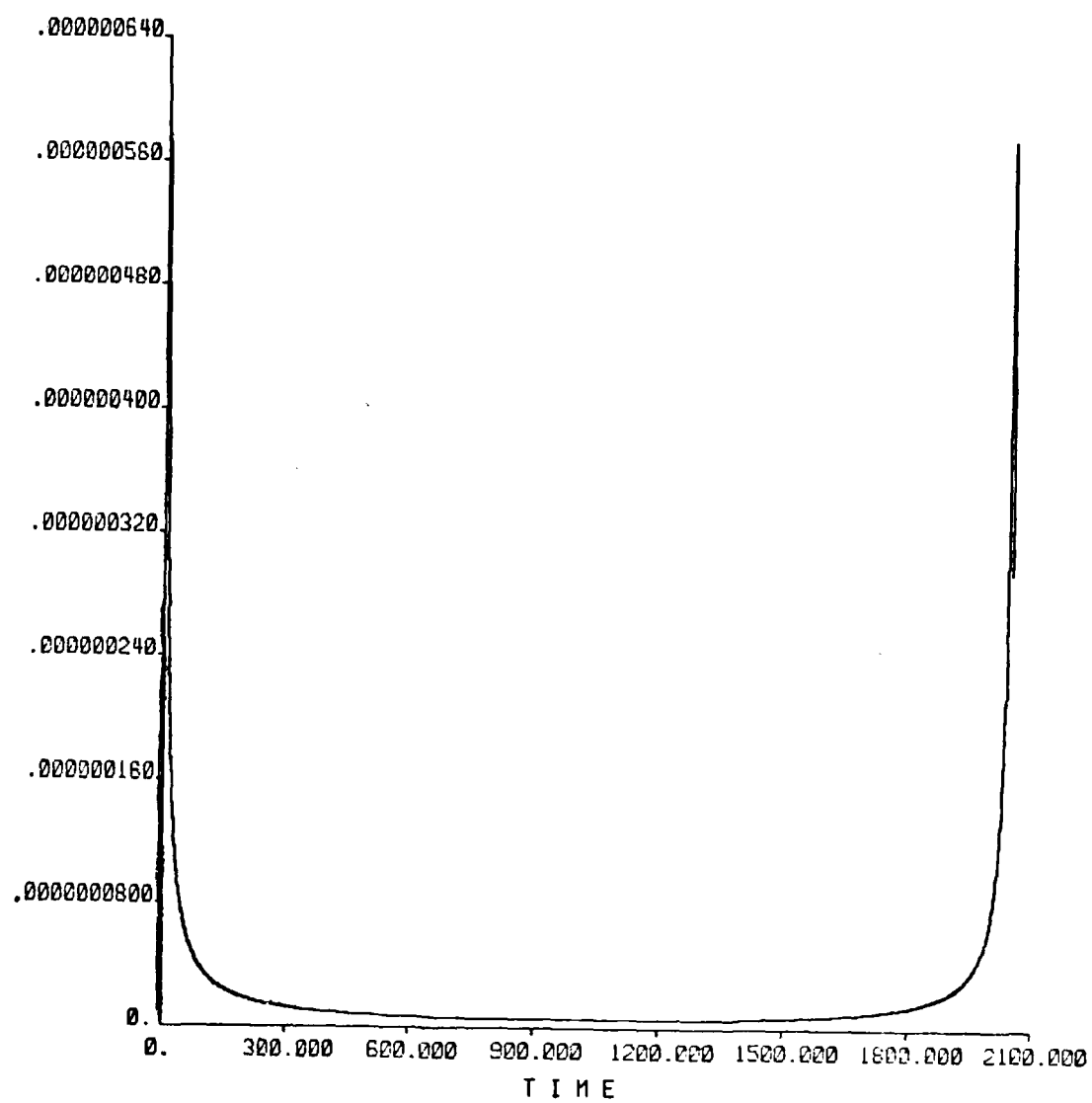


Figure 2.11.6 Magnitude of Inverse Fourier Transform of $\frac{(j\omega)^9}{9!}$

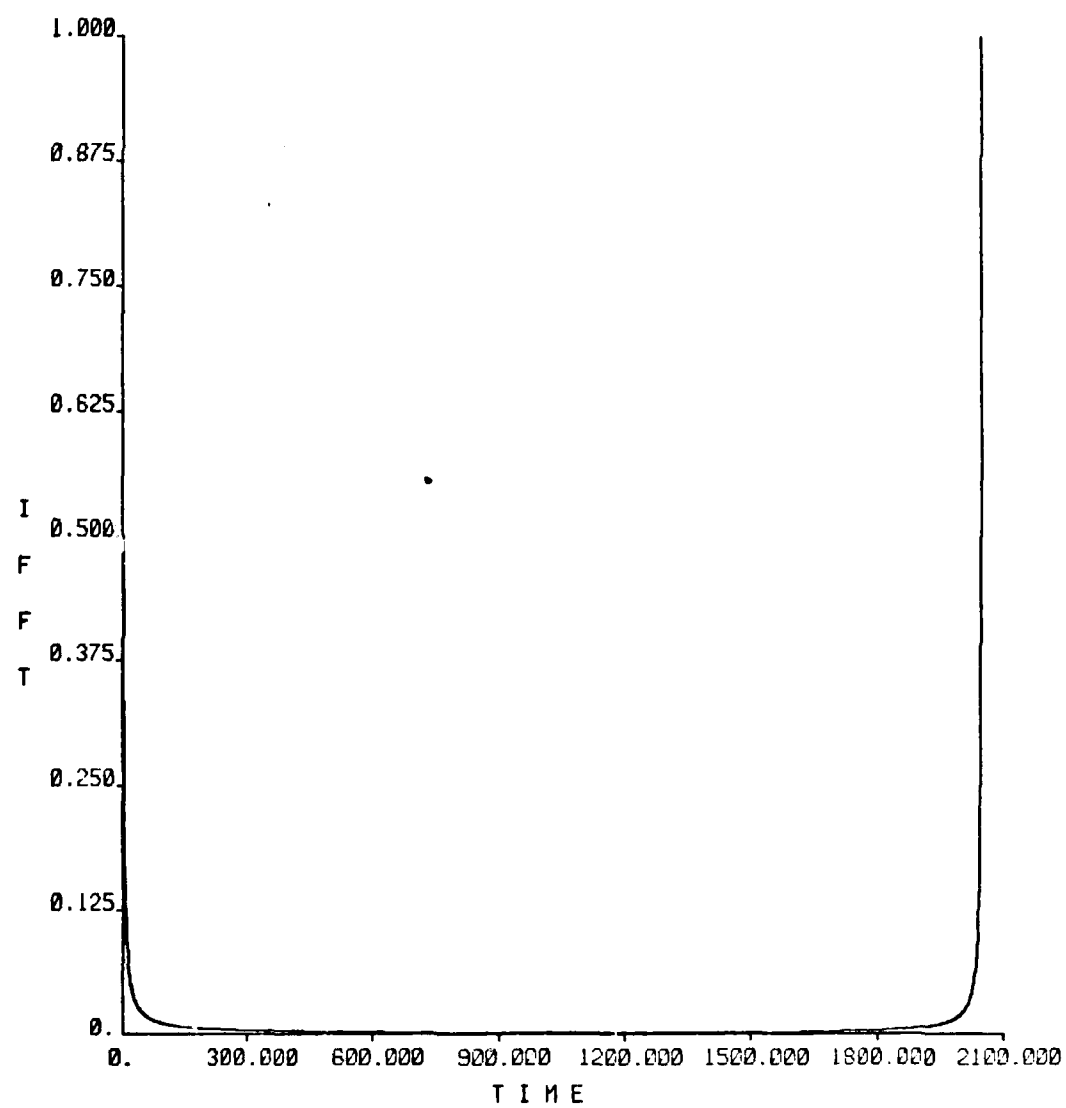


Figure 2.11.7 Magnitude of Inverse Transform of $(j\omega)$, $\frac{(j\omega)^5}{5!}$ and $\frac{(j\omega)^9}{9!}$

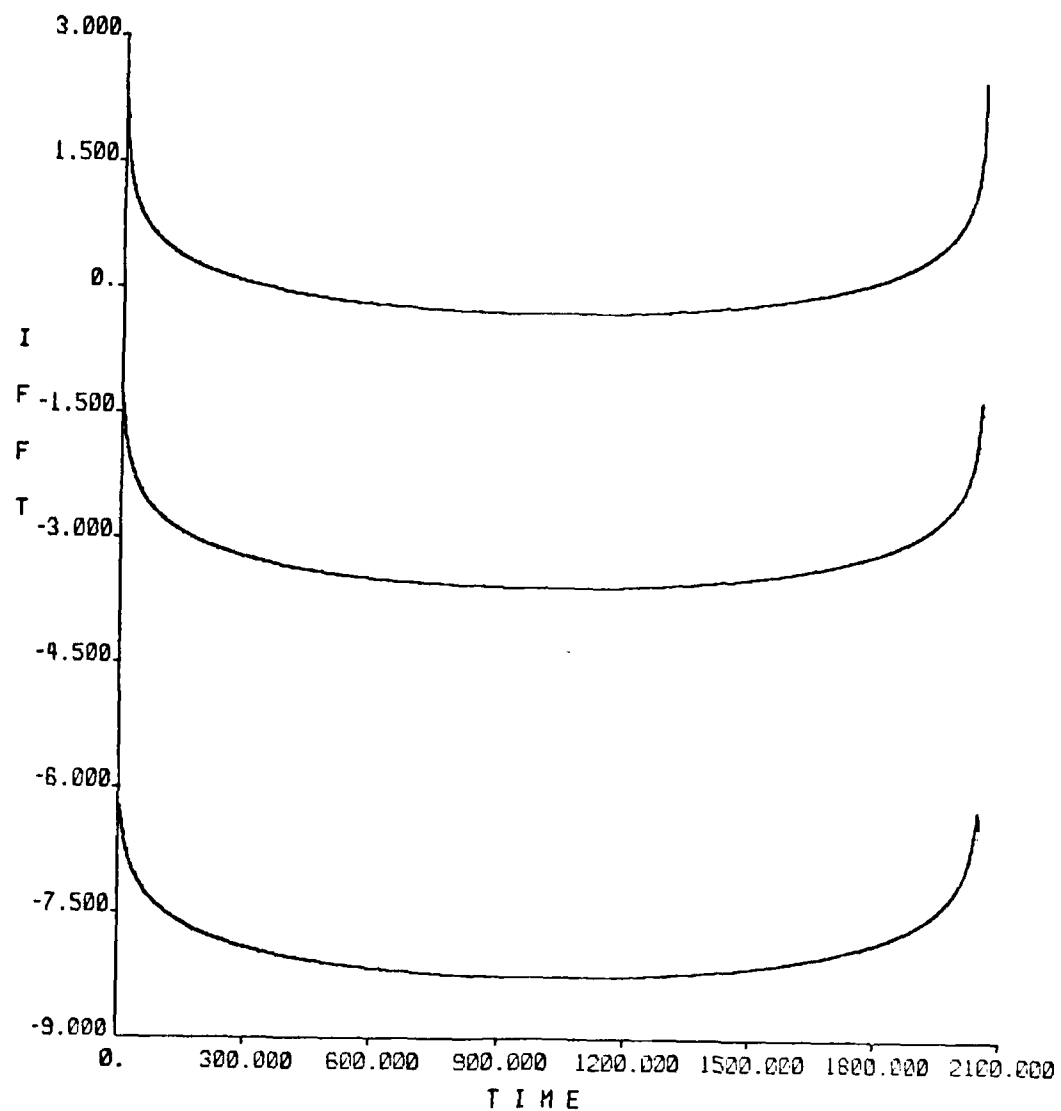


Figure 2.11.8 Log of Magnitude of Inverse Fourier Transform of $(j\omega)$, $\frac{(j\omega)^5}{5!}$, and $\frac{(j\omega)^9}{9!}$

2.12 Programming Considerations and Simulations

In this section, several technical aspects of numerically calculating the mathematics developed in this report will be considered. The four topics discussed are

- 1) cepstral calculation considerations;
- 2) cepstral errors of non-Gaussian pulses;
- 3) signal time averaging calculations;
- 4) accuracy improvement by iteration.

The definition of the cepstrum is the inverse fourier transform of the complex log of the fourier transform of the original function. Here both the forward and inverse transforms are uniquely defined but the complex log is nonunique due to the infinite number of possibilities for the imaginary term. For a complex number

$$X(t) = x_r(t) + jx_i(t), \quad (2.12.1)$$

the real part of the complex log of $X(t)$, $\hat{x}_r(t)$, is $\log \sqrt{x_r^2(t) + x_i^2(t)}$ but the imaginary part, $\hat{x}_i(t)$, is not simply the principal part of the angle of $X(t)$. It has been shown [6] that for real time signals leading to a fourier transform whose real part, $X_r(e^{j\omega})$, is even and imaginary part, $X_i(e^{j\omega})$, is odd then

$$\frac{d\hat{x}_i(e^{j\omega})}{d\omega} = \frac{\hat{x}_r^2(e^{j\omega})}{\hat{x}_i^2(e^{j\omega}) + \hat{x}_r^2(e^{j\omega})} \frac{d}{d\omega} \left[\frac{\hat{x}_i(e^{j\omega})}{\hat{x}_r(e^{j\omega})} \right] \quad (2.12.2)$$

and

$$\left| \hat{x}_i(e^{j\omega}) \right|_{\omega=0} = 0 \quad (2.12.3)$$

must be satisfied where $\hat{X}(e^{j\omega}) = \log X(e^{j\omega})$. This is satisfied when

$$\hat{X}_1(e^{j\omega}) = \int_0^\omega \arg'[X(e^{j\eta})] d\eta \quad (2.12.4)$$

and

$$\arg[X(e^{j\omega})]_{\omega=0} = 0 \quad (2.12.5)$$

This solution cannot be solved exactly numerically due to the integration but several good algorithms do exist and good results were obtained using Tribolet's program [7].

It is these complications in the complex log that will hinder the process of calculating the complete cepstrum totally within an array processor, a dedicated peripheral co-processor for array arithmetic and FFT calculations, in a one-step operation. It is possible to do the entire operation within the array processor and economical to do so in the long run, however specialized software will be needed to implement the phase unwrapping routine of [7].

The second topic dealing with noise impulses that are not Gaussian in shape was handled with some simulations. The concern here is the error introduced in the calculation of the cepstrum of the reverberant medium's transfer function due to a noise pulse whose cepstral energy is not limited to an impulse at $\tau = 0$. An ideal Gaussian pulse has a cepstrum of the form $\alpha\delta(\tau)$ and only affects the $\tau = 0$ portion of the transfer function cepstrum. Simulations were done using several pulse shapes of the form $e^{-|\tau|^n/\epsilon}$. For n greater than or equal to 2 the cepstra were impulses at $\tau = 0$. With $n = 1$ the results were not as ideal but still very promising.

Figures 2.12.1, 2.12.2 and 2.12.3 show a 512 point signal of $e^{-|t|/8}$, its fourier transform, and its cepstrum respectively. This cepstrum shows a small error introduced for very small value of τ . An improvement of this small error is shown in Figures 2.12.4, 2.12.5 and 2.12.6, which are the same respective functions but using 2048 sample points instead of 512. Again the error can be reduced by using more sample points of the collected waveform data.

Figure 2.12.4 shows the noise pulse centered at zero for convenience of calculations and to reduce the numerical error in calculating the phase which is zero for a pulse at $t = 0$. The time plot shows the other half of the pulse around $t = 2048$ due to the FFT program employed in the simulations. A second series was simulated for a shifted noise pulse. Figures 2.12.7, 2.12.9 and 2.12.10 show these results. Here the pulse was centered at 64 and a time, frequency and cepstrum plot are shown respectively. The Tribolet phase unwrapping program removes all constant phases in its calculation of the cepstrum. This constant is measured and recorded and then the cepstrum is calculated for the signal with this constant phase removed. For plotting purposes, this constant is easily reintroduced with the cepstrum by a simple shift of the values by the amount of the constant. In this case the constant is always equal to the time delay and the cepstrum shows a spike at 64, the center point of the delay pulse. Also to be noted here is that the phase in 2.12.9, and later in 2.12.13, has not been unwrapped.

Figures 2.12.12, 2.12.13 and 2.12.14 show a time signal with two pulses, $e^{-|t-32|/8}$ and $.5e^{-|t-160|/8}$, its fourier transform and its cepstrum. The second pulse has been attenuated by .5.

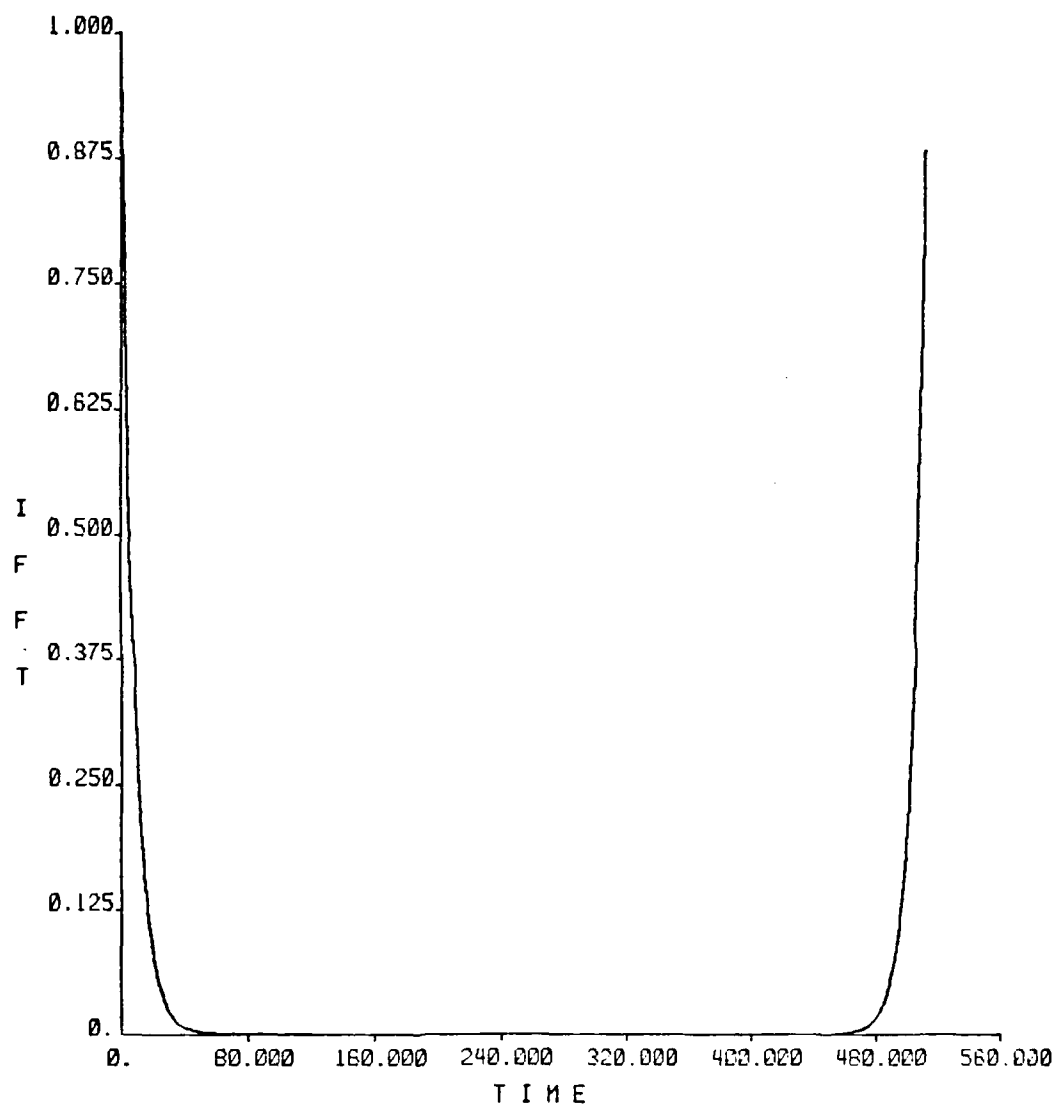


Figure 2.12.1 Noise Pulse, $x(t) = e^{-|t|/8}$; 512 points

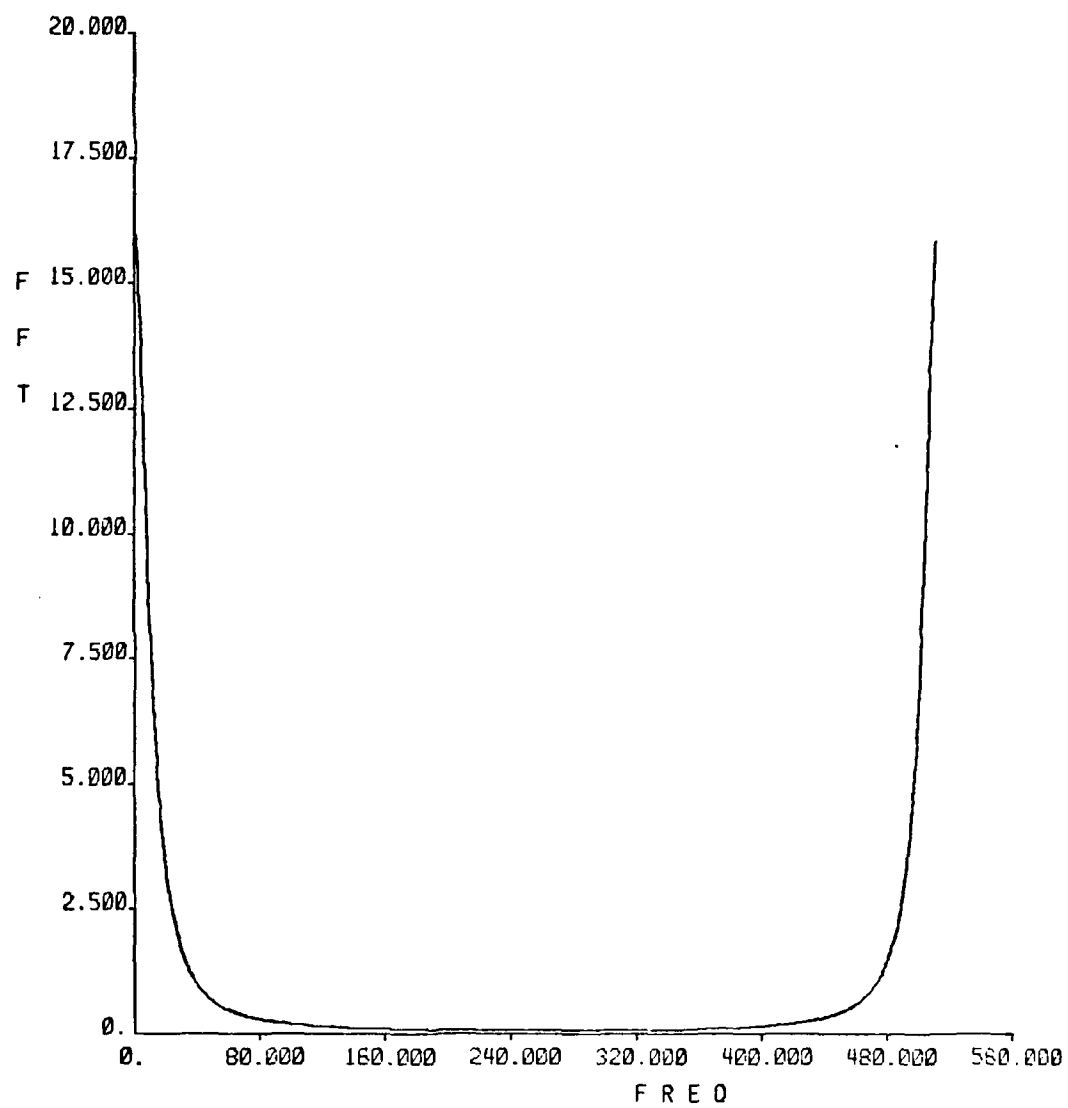


Figure 2.12.2 Fourier Transform of Noise Pulse, $F[e^{-|t|/8}]$;
512 points

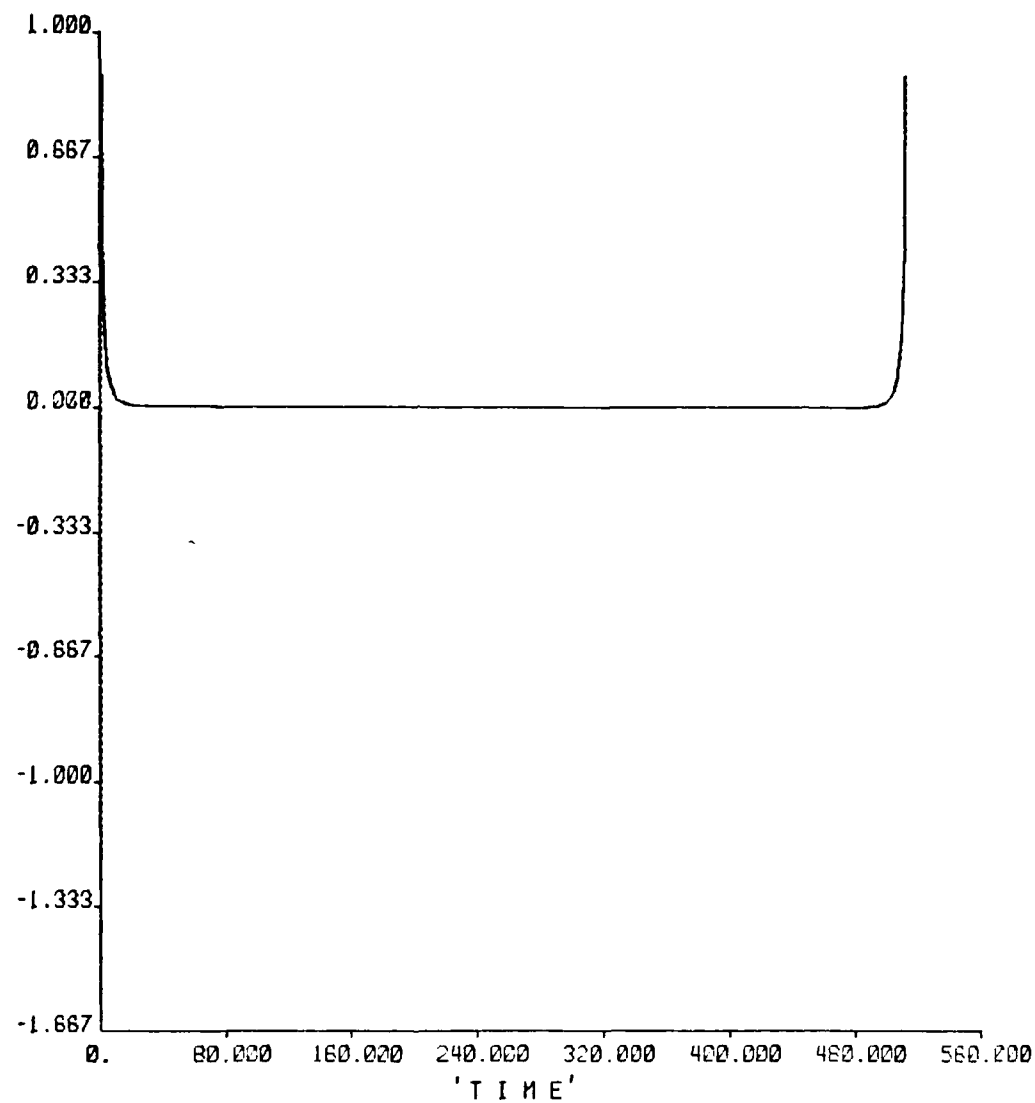


Figure 2.12.3 Cepstrum of Noise Pulse; 512 points

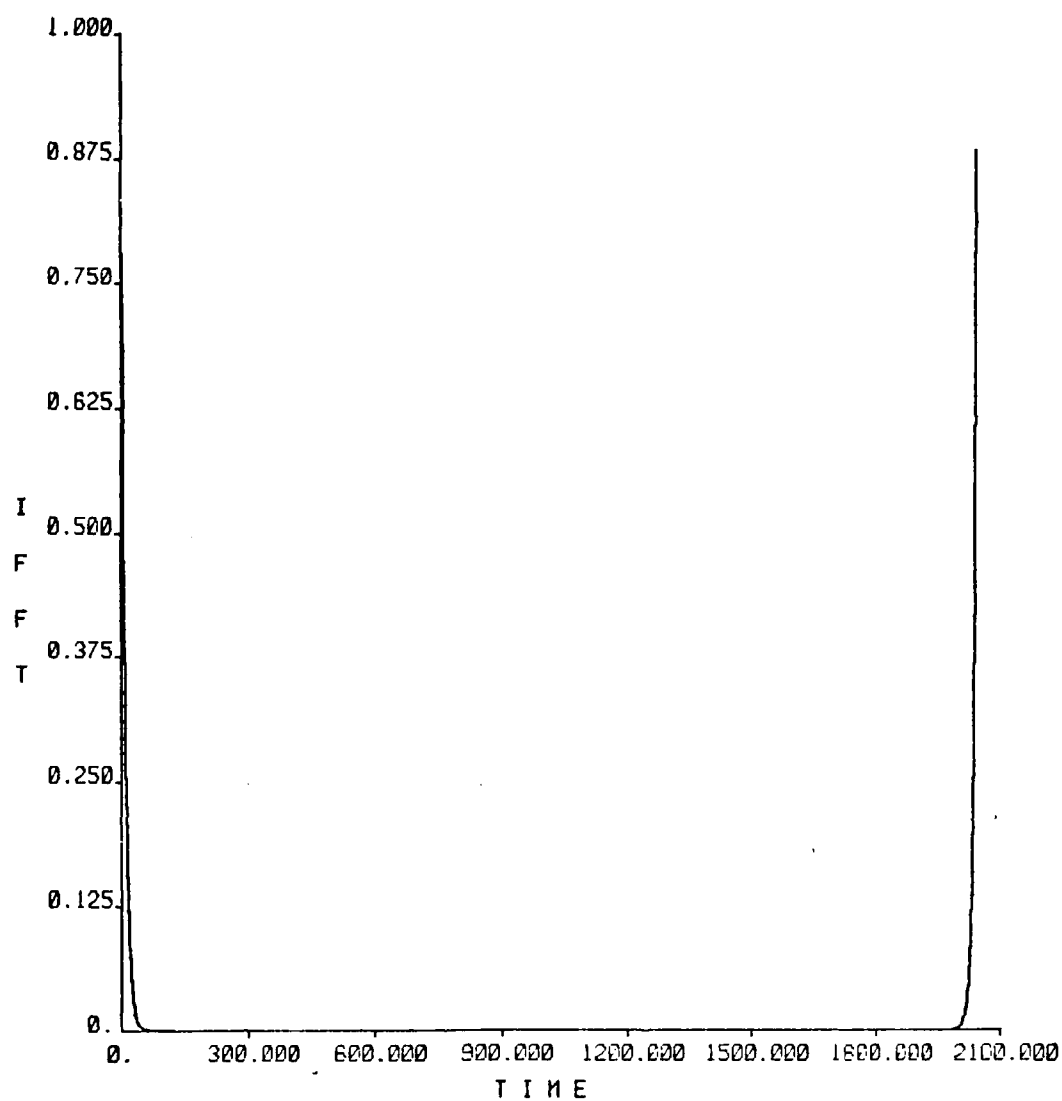


Figure 2.12.4 Noise Pulse, $x(t) = e^{-|t|/8}$; 2048 points

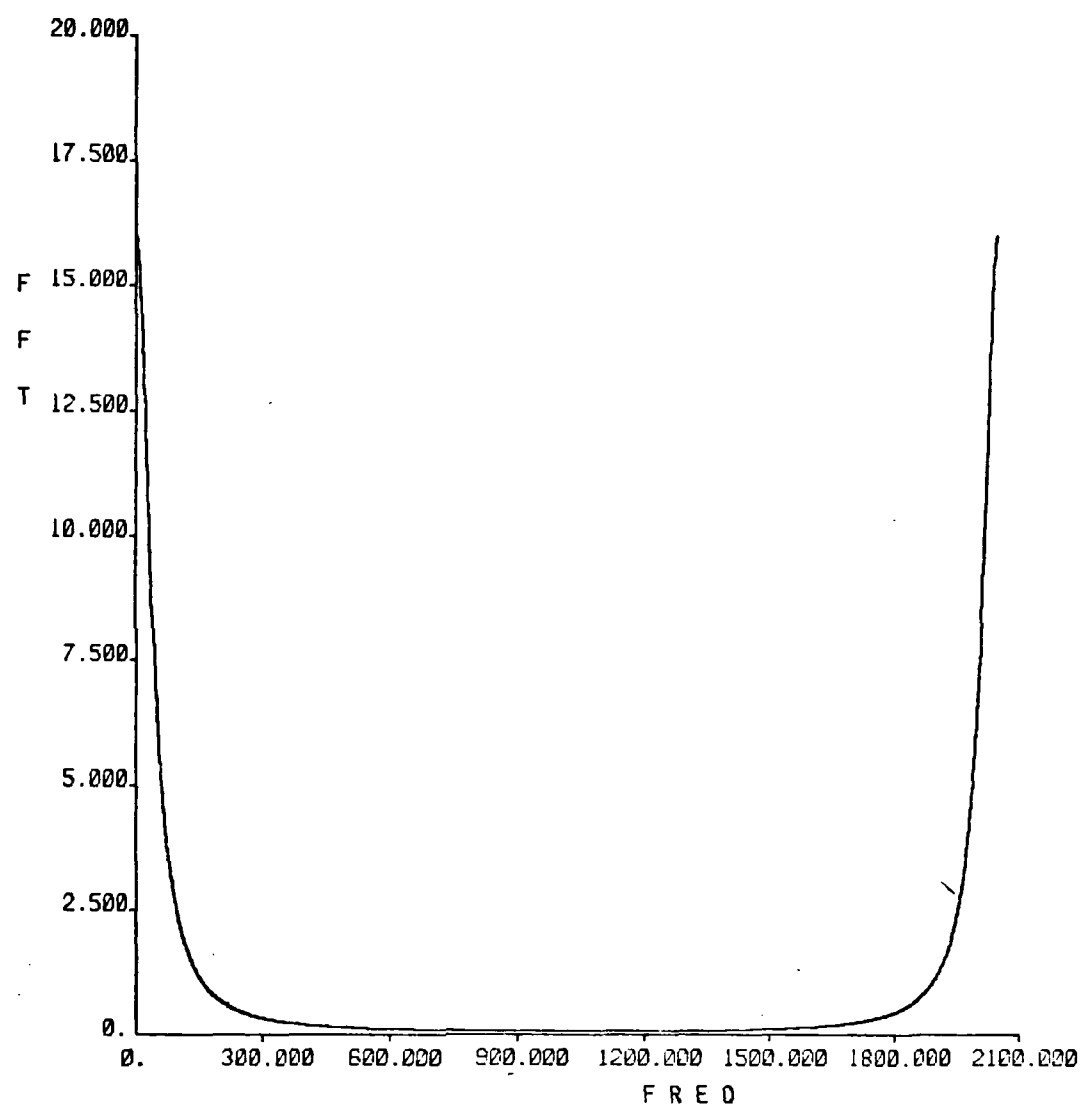


Figure 2.12.5 Fourier Transform of Noise Pulse, $F[e^{-|\tau|/8}]$
2048 points

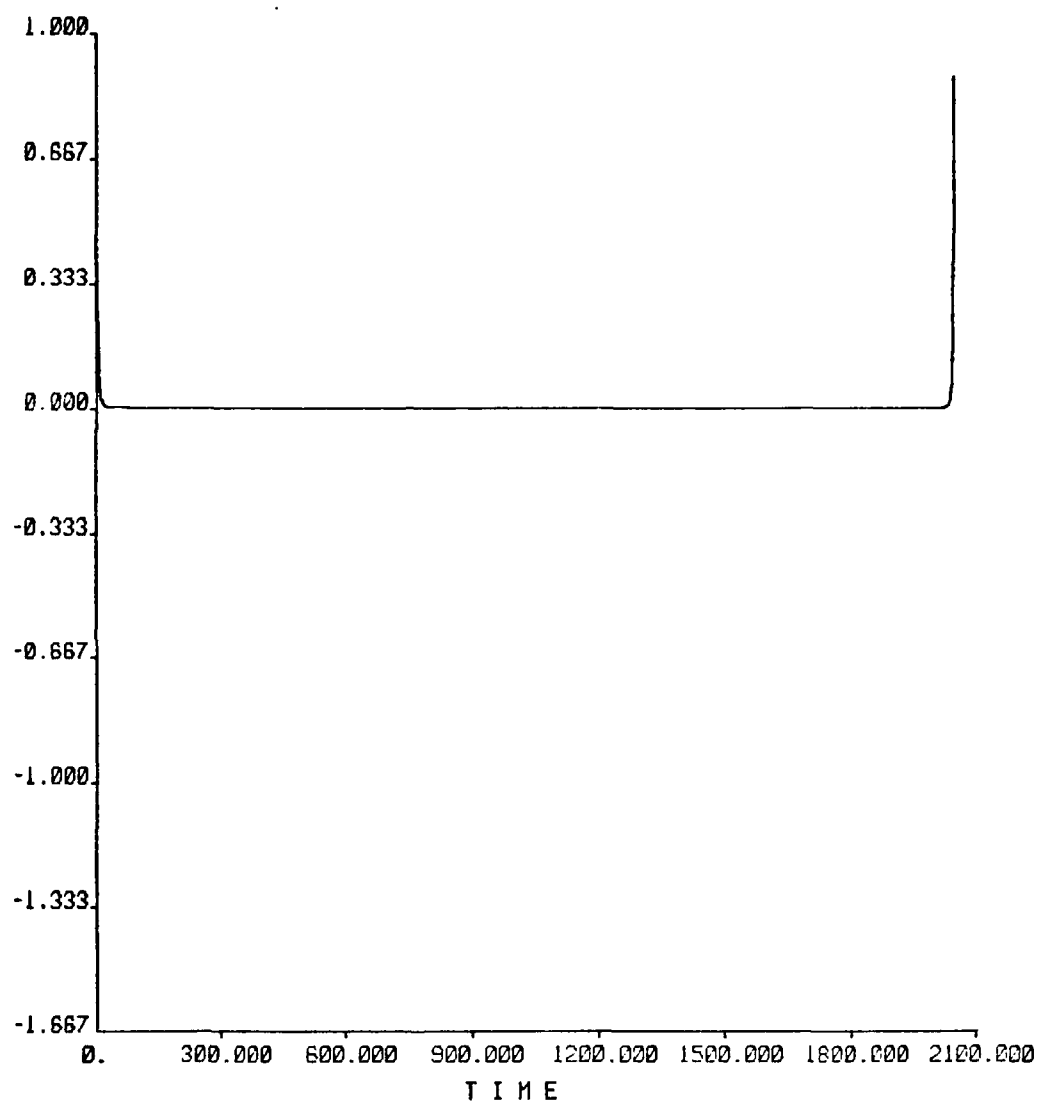


Figure 2.12.6 Cepstrum of Noise Pulse; 2048 points

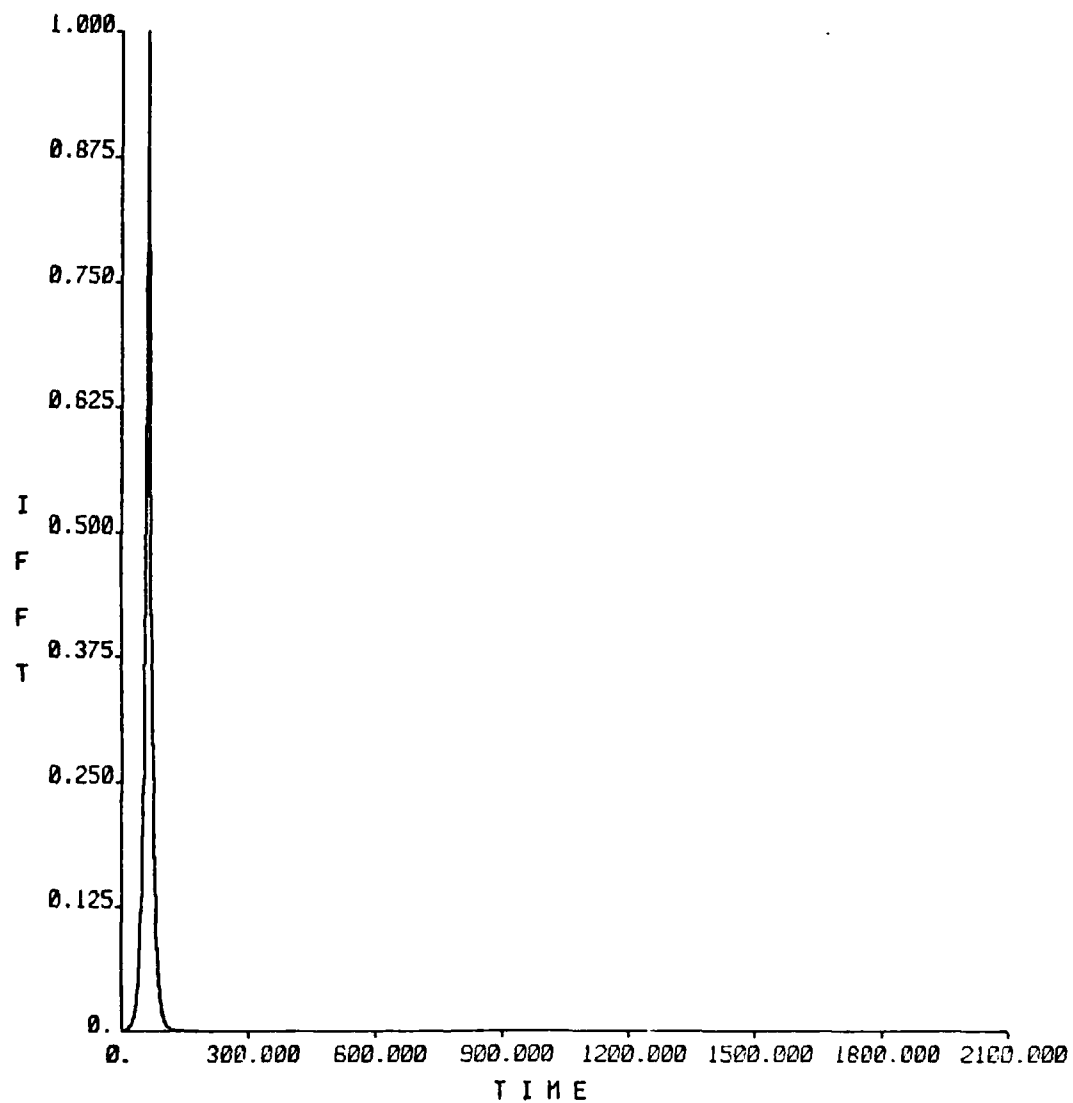


Figure 2.12.7 Delayed Noise Pulse $e^{-|t-64|/8}$; 2048 points

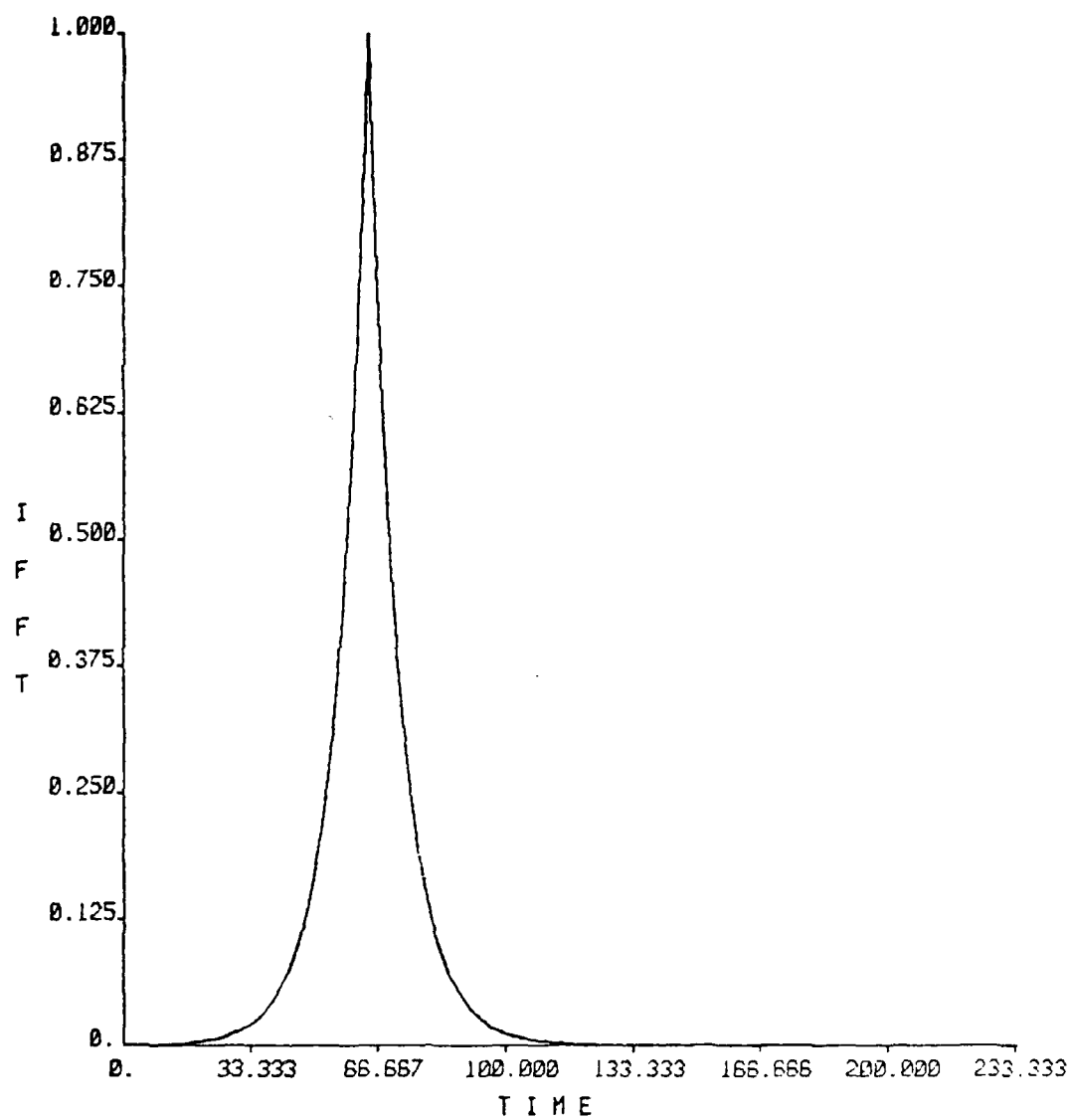


Figure 2.12.8 Blow-up of Noise Pulse of Figure 2.12.7

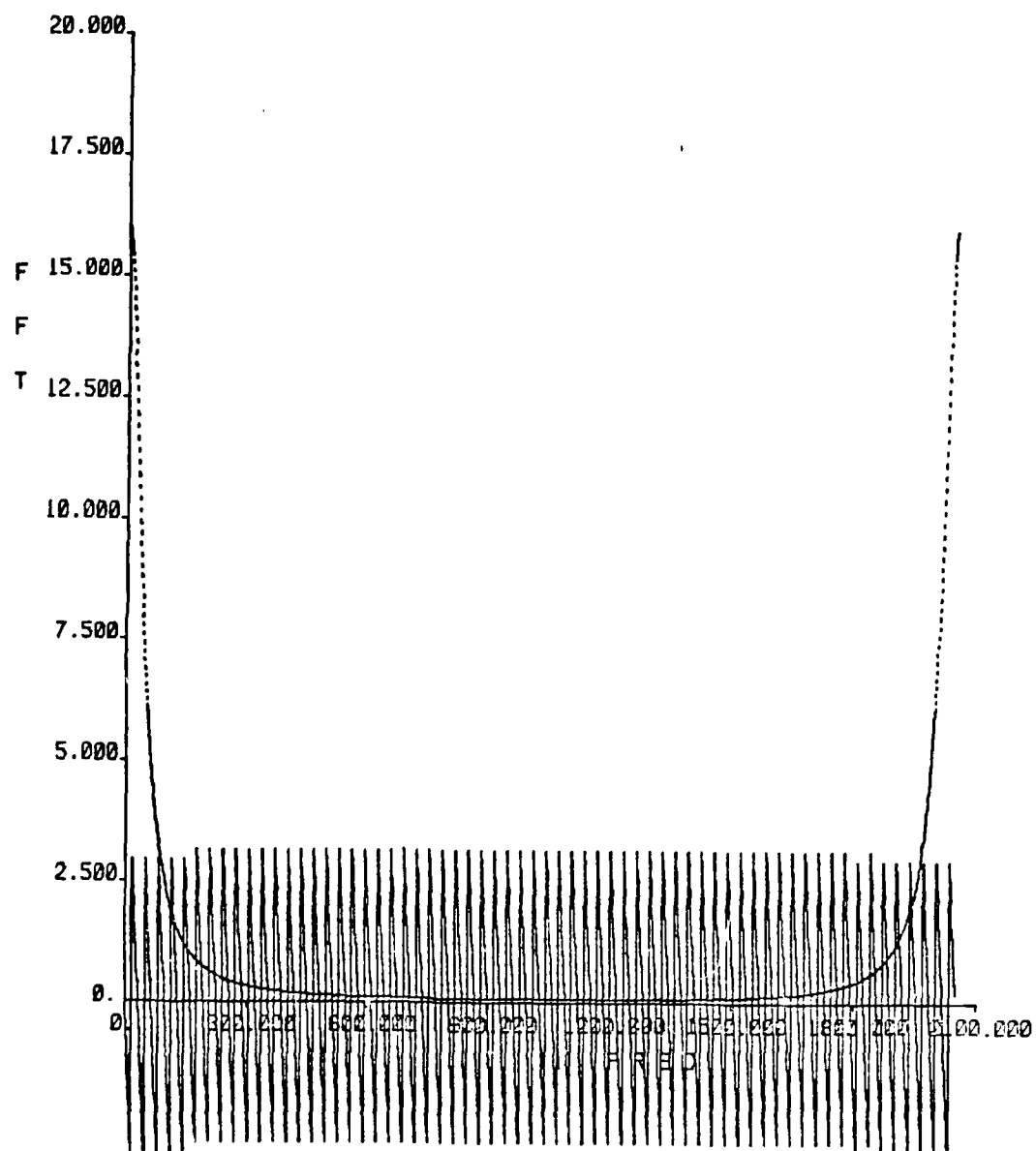


Figure 2.12.9 Fourier Transform of Delayed Noise Pulse; 2048 points. Dotted line is magnitude, solid line is phase.

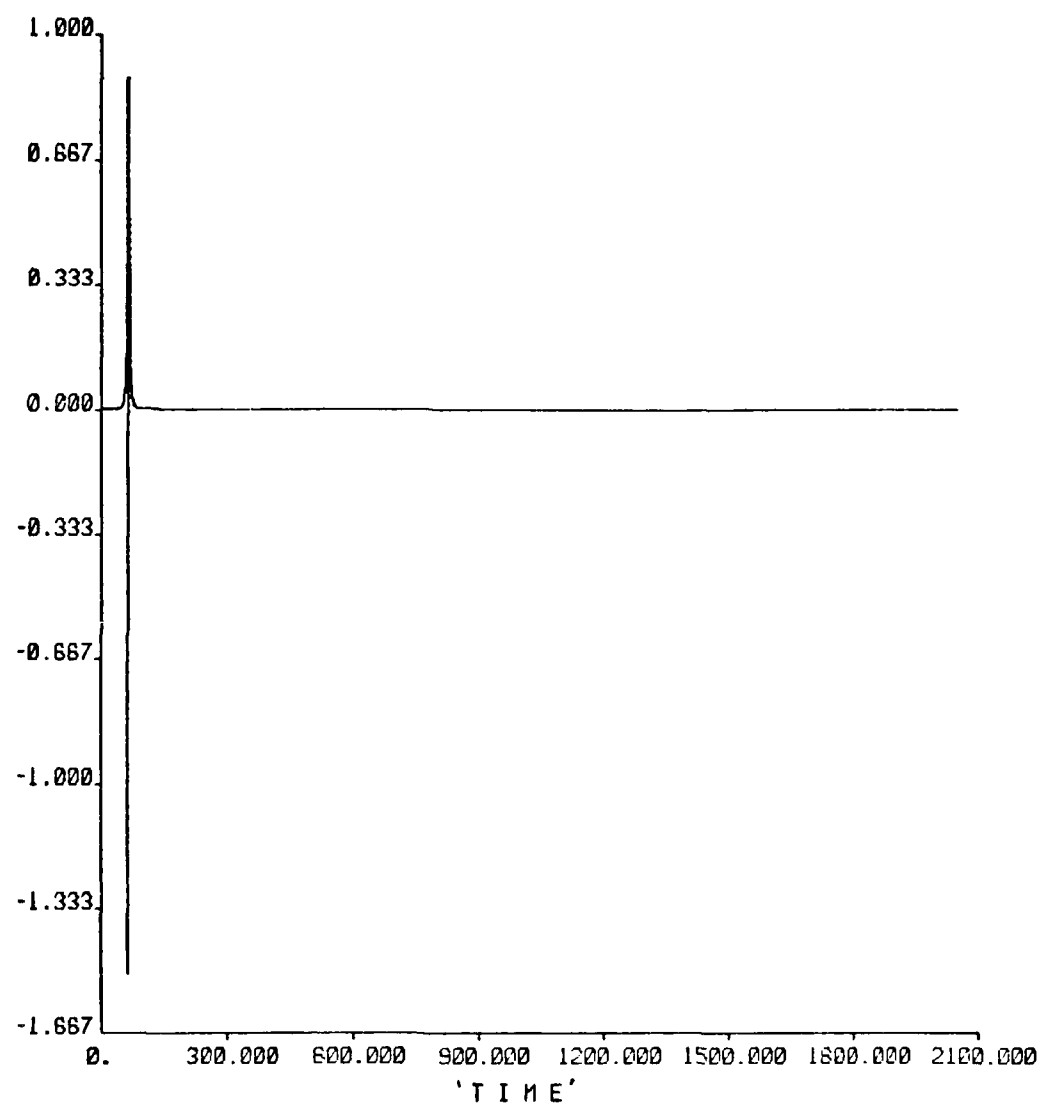


Figure 2.12.10 Cepstrum of Delayed Noise Pulse;
2048 points

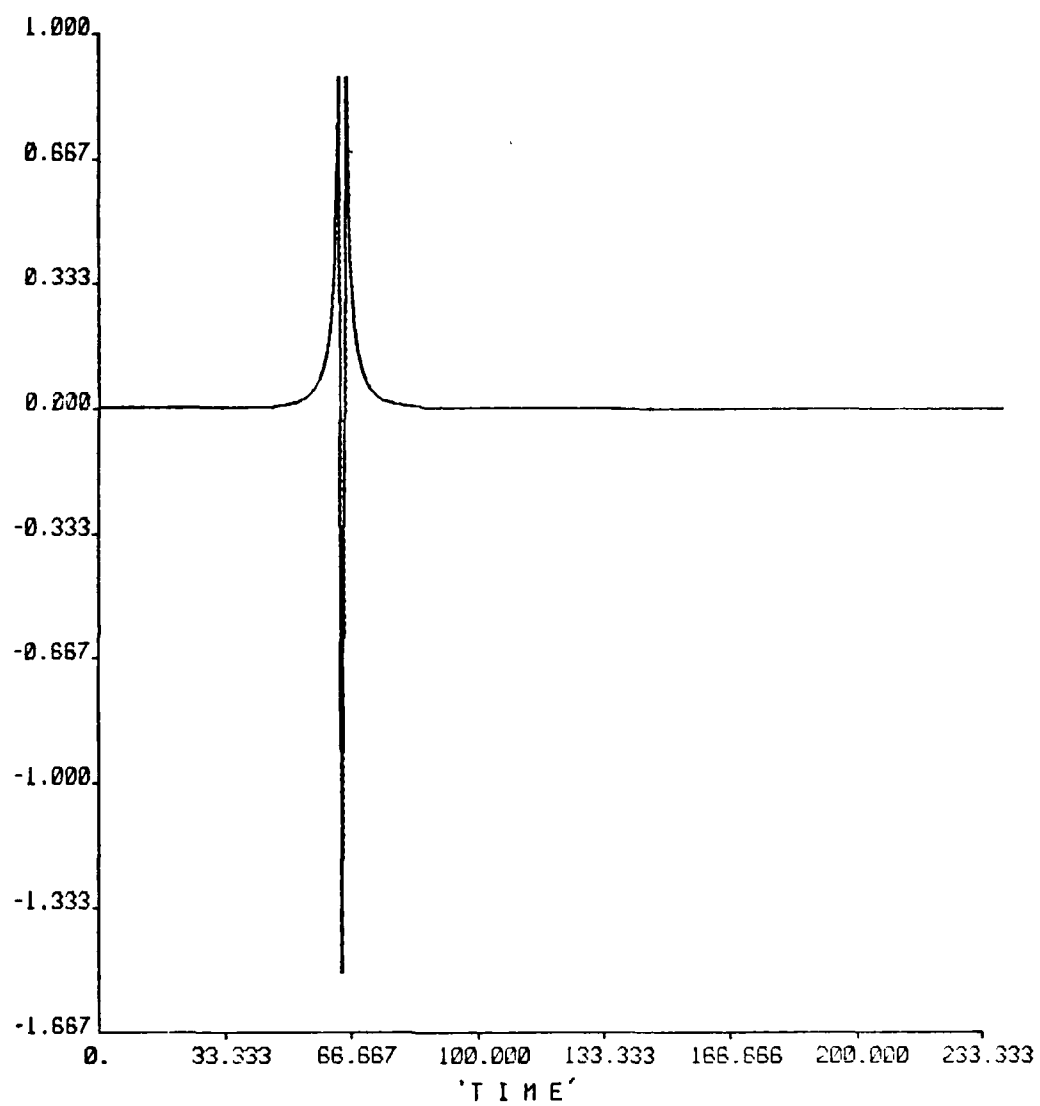


Figure 2.12.11 Blow-up of Cepstrum of Figure 2.12.10

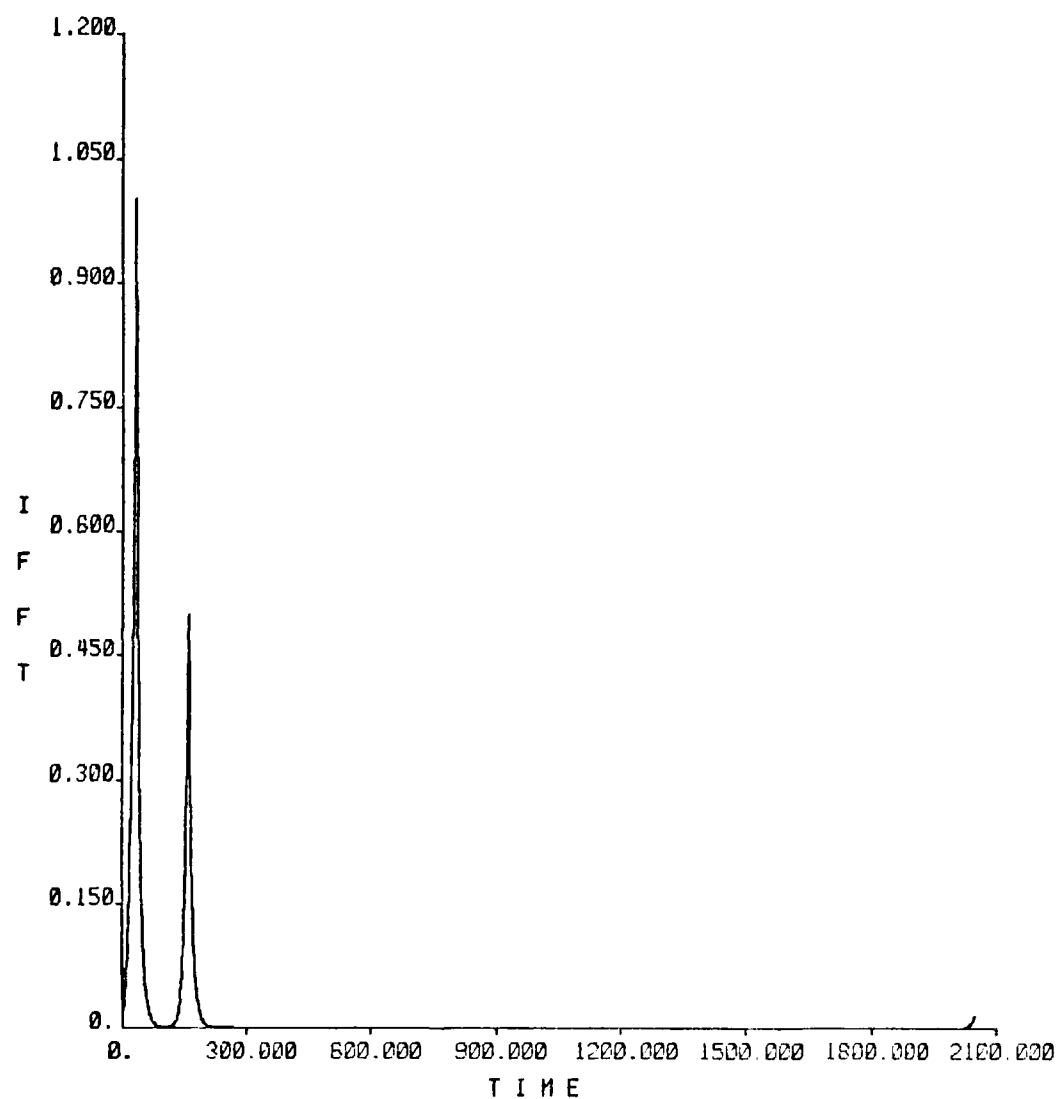


Figure 2.12.12 Dual Noise Pulses, $e^{-|t-32|/8} + .5 e^{-|t-160|/8}$;
2048 points

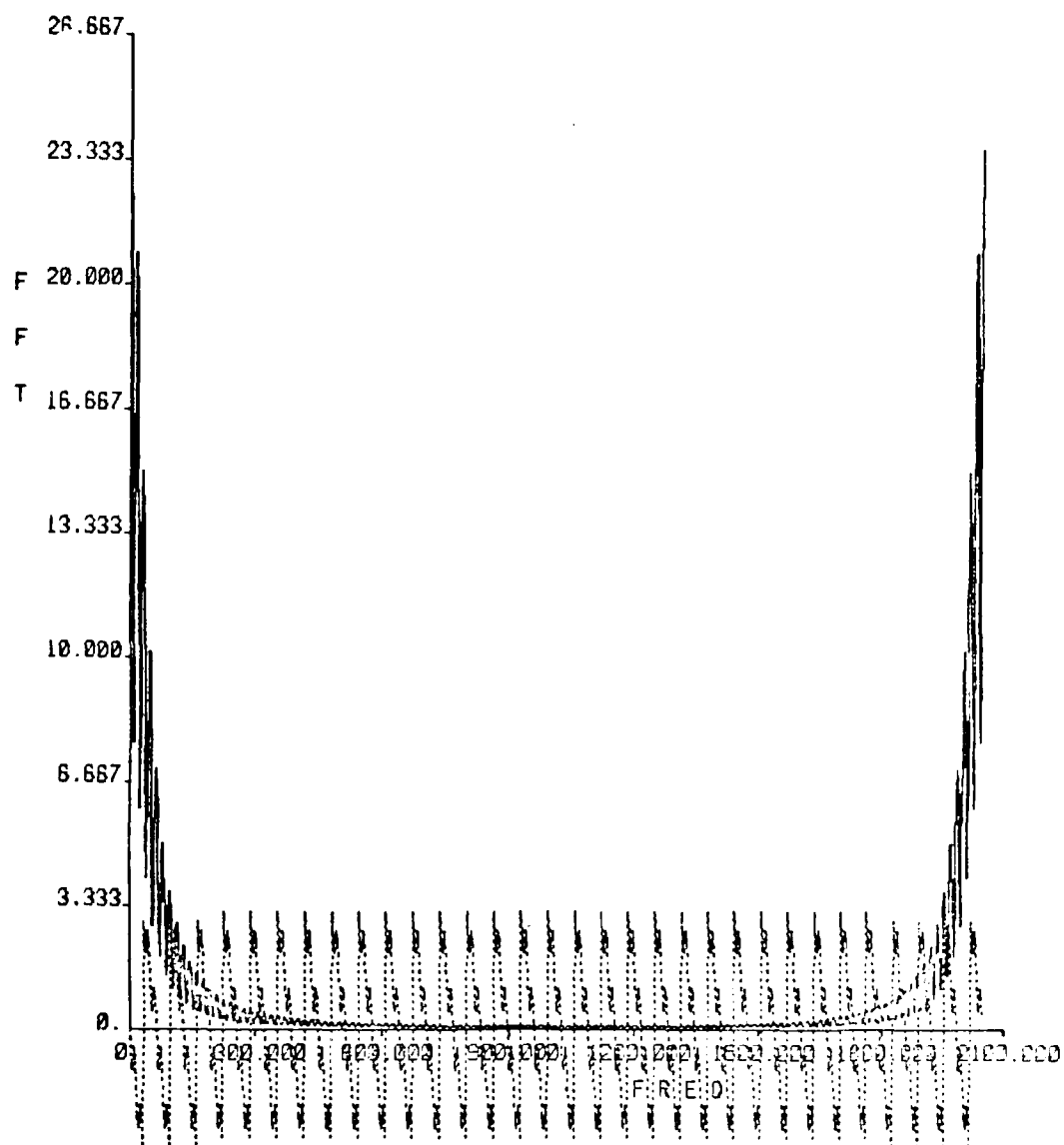


Figure 2.12.13 Fourier Transform of Dual Noise Pulses;
2048 points

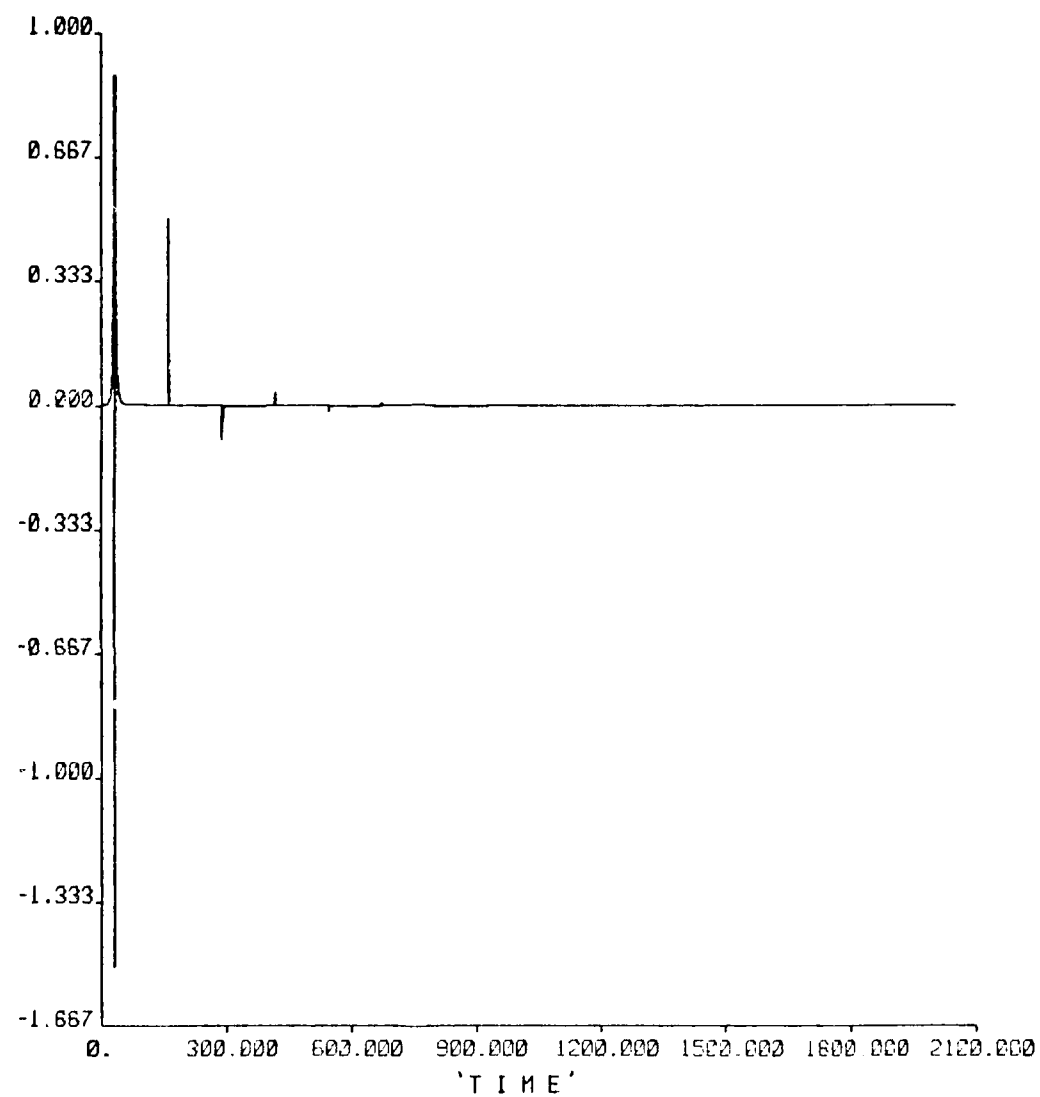


Figure 2.12.14 Cepstrum of Dual Noise Pulses; 2048 points

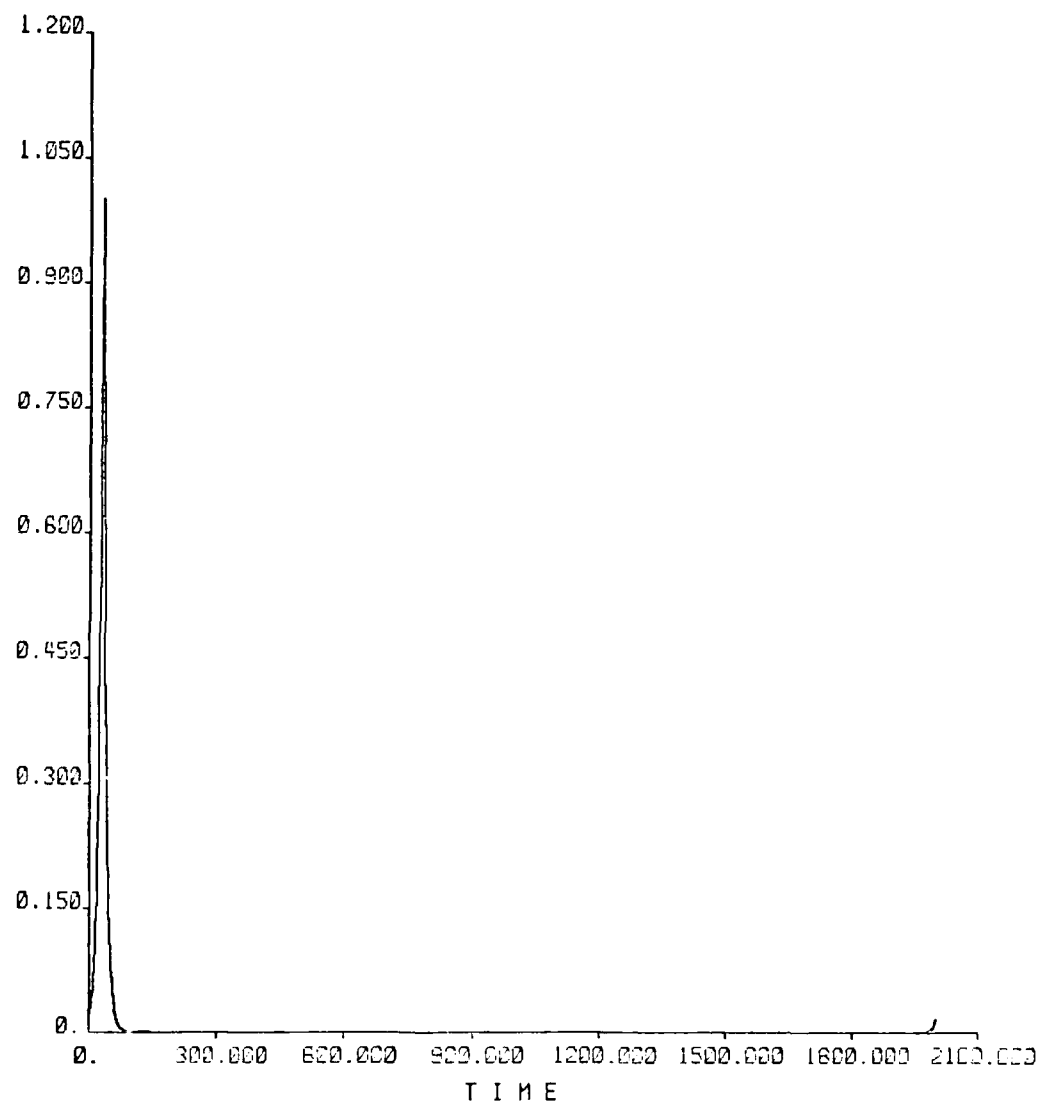


Figure 2.12.15 Normalized Deresonated Noise Pulse
(Magnitude): 2048 points

The pulse in the cepstrum due to the pulse shape appears at $\tau = 0$ as expected. A series of decaying pulses also appear in the cepstrum which are due to "harmonics" of the integral multiples of the time difference between the two pulses. The doublet functions found in all the cepstra can be related to the $j\omega$ term and the pulse shape term which appears after the logging operation. This transforms to a derivative operation on the impulses in the cepstral domain added to a broadened spike function due to the pulse shape.

Using the dual pulse signal of Figures 2.12.12, 2.12.13 and 2.12.14, the general results can be verified mathematically. The time signal is of the form

$$x(t) = e^{-|t-\tau_1|} + ae^{-|t-\tau_2|} \quad (2.12.6)$$

Its fourier transform is

$$X(\omega) = 2 \frac{e^{j\omega\tau_1}}{1 + \omega^2} + 2a \frac{e^{j\omega\tau_2}}{1 + \omega^2} \quad (2.12.7)$$

$$= \frac{2e^{j\omega\tau_1}}{1 + \omega^2} [1 + ae^{j\omega(\tau_2 - \tau_1)}] \quad (2.12.8)$$

Taking the log and using the expansion of eq. (2.5.12) gives

$$\log X(\omega) = j\omega\tau_1 + \log 2 - \log(1 + \omega^2) + \log[1 + ae^{j\omega(\tau_2 - \tau_1)}] \quad (2.12.9)$$

$$= j\omega\tau_1 + \log 2 - \log(1 + \omega^2) - \sum_{i=1}^{\infty} \frac{[-ae^{-j\omega(\tau_2 - \tau_1)}]^i}{i} \quad (2.12.10)$$

and taking the inverse fourier transform yields

$$X(\tau) = \tau_1 \frac{d}{d\tau} \delta(\tau) + \log(2) \delta(\tau) + P(\tau) - \sum_{i=1}^{\infty} \frac{(-a)^i}{i} \delta[i(\tau_2 - \tau_1)] \quad (2.12.11)$$

$$\begin{aligned}
&= \tau_1 \frac{d}{dt} \delta(\tau) + \log(2) \delta(\tau) + (\tau) + a \delta(\tau_2 - \tau_1) - \frac{a^2}{2} \delta(2(\tau_2 - \tau_1)) \\
&\quad + \frac{a^3}{3} \delta(3(\tau_2 - \tau_1)) - \dots
\end{aligned} \tag{2.12.12}$$

where

$$P(\tau) = F^{-1}\{-\log(1 + \omega^2)\} \tag{2.12.13}$$

$P(\tau)$ is not easily evaluated but it can be seen to be a narrow pulse around $\tau = 0$, since $1/(1 + \omega^2)$ is a narrow pulse whose cepstrum will be even sharper. The last term in Eq. (2.11.11) is a series of alternating signs decaying impulses (assuming a is less than 1). This is verified by Figure 2.12.14. In this figure, the constant delay factor of τ_1 has been reinserted so that the initial spike occurs where the first time pulse occurs. As seen in the figure and the equations, the third pulse in the cepstrum which always occurs at $2(\tau_2 - \tau_1)$ may not always be ignored in dereverberation calculations.

A few comments should be made here about the Tribolet phase unwrapping program. In general, it seems to have problems with ideal functions and takes much longer to compute the cepstra of noiseless numerical data. Also as noted in Tribolet's presentation, there are two values which can be adjusted to improve the accuracy of the routine; however this will substantially increase the computation time. Occasionally some very irregular oscillating pulses were obtained in the cepstra with some input pulses. Finally, although not used in these simulations, all calculations should be done in double precision as was suggested by Tribolet for improved accuracy and better results.

The problem of deciding how many waveforms must be averaged to obtain a good estimate of the necessary time averages used in the mathematics developed earlier is indeed a difficult one. The necessary waveforms are hard to simulate for later use in actual calculations for the purpose of coming up with some working estimates of numbers needed for reasonable accuracy. Secondly, little is known about the characteristics of these waves for the purpose of calculating reasonable numbers to be averaged for good results. These parameters can best be found by trial and error with actual data while closely monitoring the intermediate results and comparing them to the mathematical ideals.

Finally some comments will be made here about numerically evaluating results for equations presented in Section 3.2 of this report. An iterative method is developed for improving the accuracy of some calculated results. Iterations are suggested to be continued until little change is observed from one step to the next, at which point it is assumed that the results are sufficiently accurate. Since these iterations may be costly and time-consuming, some experimentation should be done comparing the end results of the completed iterations to several partially completed iterations and determine a lower limit on the number of loops necessary for good results. Also, an obvious check should be included in the program to determine the possibility of divergence in some situations. These iterations might be done completely in an array processor but would certainly require much specialized software.

2.13 Relations Between Cepstra and Predictor Coefficients

Alternate processing of signals in speech and seismic studies frequently makes use of adaptive predictors to establish signal structure. One might wonder whether such processing would be of value in the context of the dereverberation problem. In this section some existing literature is reviewed which shows that a subtle relation exists between predictive processing and cepstral processing. In the course of making this connection it will be shown that it makes use of a relation (2.13.19) which was found to be useful in earlier studies.

Adaptive processing [11] can be employed to identify certain linear systems whose driving source is not accessible; that is, the source cannot be chosen and its value is not known. Adaptive processing in this case can only operate on the observed output of the system or process. One approach consists of forcing a linear predictor to predict a future system output based on only the past observed system outputs. In forcing the predictor to adapt to minimize the error of prediction, a great deal is learned about the process structure.

In the case of the single source dereverberation problem of Section 2.4, this section demonstrates a means of obtaining the cepstrum of the received signal for use in finding the channel response by Equation (2.4.5). If the p predictor coefficients b_k , $k = 1, 2, \dots, p$ of $\tilde{y}(t)$, the average time signal received, are known then the discrete cepstral coefficients c_n can be found either directly or recursively.

An optimum set of predictor coefficients can be found for a given signal [81] in terms of the signal's autocorrelation function and variance. Given an input signal $y(t)$ or in discrete form $y(n)$ with a predicted value $\tilde{y}(n)$ then the prediction error signal is

$$d(n) = y(n) - \tilde{y}(n) \quad (2.13.1)$$

Since

$$\tilde{x}(n) = \sum_{k=1}^p b_k x(n-k) \quad (2.13.2)$$

the optimal predictor coefficients, b_k 's, are the ones which minimize

$$\sigma_d^2 = E\{d^2(n)\} = E\{[x(n) - \tilde{x}(n)]^2\} \quad (2.13.3)$$

$$= E\{[x(n) - \sum_{k=1}^p b_k x(n-k)]^2\} \quad (2.13.4)$$

They are found by setting the partial derivatives of σ_d^2 with respect to the b_k 's to zero, i.e.,

$$\frac{\partial \sigma_d^2}{\partial b_j} = -2E\{[x(n) - \sum_{k=1}^p b_k x(n-k)] x(n-j)\} = 0 \quad ; \quad 1 \leq j \leq p \quad (2.13.5)$$

This can be rewritten as

$$E\{[x(n) - \tilde{x}(n)] x(n-j)\} = E\{d(n) x(n-j)\} = 0 \quad ; \quad 1 \leq j \leq p \quad (2.13.6)$$

or

$$E\{x(n-j) x(n)\} = \sum_{k=1}^p b_k E\{x(n-j) x(n-k)\} \quad ; \quad 1 \leq j \leq p \quad (2.13.7)$$

Defining $\phi(j)$ as the autocorrelation function of $x(n)$ then (2.13.7) becomes

$$\phi(j) = \sum_{k=1}^p b_k \phi(j-k) \quad (2.13.8)$$

which can be written in matrix notation as

$$\bar{\rho} = \bar{C}\bar{b} \quad (2.13.9)$$

when

$$\rho(j) = \frac{\phi(j)}{\sigma_x^2} \quad (2.13.10)$$

and

$$\bar{C} = \begin{bmatrix} 1 & \rho(1) & \dots & \rho(p-1) \\ \rho(1) & 1 & \dots & \rho(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \dots & 1 \end{bmatrix} \quad (2.13.11)$$

The b_k 's can be found from

$$\bar{b} = \bar{C}^{-1}\bar{\rho} \quad (2.13.12)$$

Hardware can be built to carry out these functional evaluations and output the optimal predictor values automatically when an appropriate signal $y(n)$ is input (See Figure 2.13.1). This equipment can also be made to run adaptively with updated coefficients returned periodically which track a time-varying (nonstationary) input signal.

Given $Y(z)$ is the Z-transform of $\tilde{y}(t)$ and also

$$Y(z) = \sum_{k=0}^p b_k z^{-k}; \quad a_0=1 \quad ; \quad a_p \neq 0 \quad (2.13.13)$$

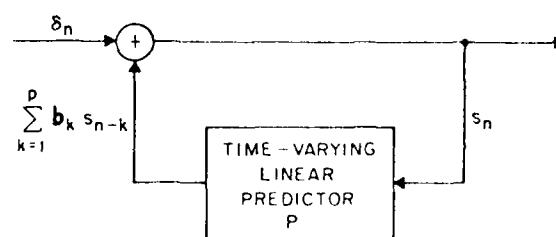


Figure 2.13.1 Adaptive Linear Prediction of Source Signal (after Atal and Hanauer)

where the p b_k 's are the least linear mean square predictor coefficients of $\tilde{y}(t)$, then the Z-transform cepstral coefficients c_n are defined as

$$\ln Y(z) = \sum_{n=1}^{\infty} c_n z^{-n} \quad (2.13.14)$$

and can be related to the b_k in several ways.

The well-known recursive relation [10] between the b_k and c_n is obtained by differentiating (2.13.14) with respect to z^{-1} and equating equal powers of z^{-1} , giving

$$c_n = b_n + 1/n \sum_{k=1}^{n-1} k c_k b_{n-k} \quad (2.13.15)$$

This is very straightforward and most useful since all of the cepstrum except the region around the origin is needed to calculate the channel response for dereverberation.

A second and non-recursive relation between the c_n and b_k was derived by applying a formula for the division of two power series to the ratio $-A'(Z)/A(Z)$ obtained after differentiating the left side of (2.13.14) [11]. This yields

$$c_n = \frac{-1}{n} (-1)^n \begin{vmatrix} b_1 & 1 & 0 & 0 & \dots & 0 \\ 2b_2 & b_1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ nb_n & b_{n-1} & b_{n-2} & \dots & b_1 & 1 \end{vmatrix} \quad (2.13.16)$$

Due to the awkwardness in evaluating this determinant another formulation for finding the c_n was derived in [11]. Using the series expansion for $\ln(1+x)$ after substituting (2.13.13) into (2.13.14) and manipulation around the summation signs yields

$$c_n = - \sum \frac{(k_1 + k_2 + \dots + k_p - 1)!}{k_1! \dots k_p!} (-b_1)^{k_1} \dots (-b_p)^{k_p} \quad (2.13.17)$$

where the sum is taken over all k_r such that

$$k_1 + 2k_2 + \dots + pk_p = n \quad (2.13.18)$$

Although the formulation of (2.13.17) is not as simple as that of (2.13.4) it is definitely more mathematically tractable and computationally efficient.

It was also shown in [11] that the inverse relationship of (2.13.17) can be found. Equation (2.13.13) and (2.13.14) yield

$$\sum_{n=0}^p b_n z^{-n} = \exp \left[\sum_{k=1}^{\infty} c_k z^{-k} \right] \quad (2.13.19)$$

Expanding the exponential function into a power series gives

$$\sum_{n=0}^p b_n z^{-n} = \sum_{m=0}^{\infty} \frac{1}{m!} \left[\sum_{k=1}^{\infty} c_k z^{-k} \right]^m \quad (2.13.20)$$

which after tedious manipulations around the summations and evaluation at appropriate powers of m yields

$$a_n = \sum \frac{(-c_1)^{k_1} \dots (-c_n)^{k_n}}{k_1! \dots k_n!} \quad (2.13.21)$$

where the sum is taken over all k_p such that

$$k_1 + 2k_2 + \dots + nk_n = n \quad (2.13.22)$$

and

$$k_1 + k_2 + \dots + k_n = m \quad (2.13.23)$$

The notation used in this section has been that of the Z-transform. However, this is just a more general form of the DFFT where $e^{j\omega T}$ is substituted for z . The results here can thus be applied to the problems stated in the other sections of this work.

In summary then it has been shown that the p coefficients b_k produced by the optimum linear predictors are related to the cepstral Z-transform coefficients c_n by several forward and reverse mapping equations which have been derived again from the relation of equation (2.13.19).

2.14 Errors Due to Estimation Noise

Fluctuations in moment estimates or other averaging functions which operate on real stochastic data add a certain amount of noise to the results. Increasing the number of signals used in these averaging processes reduces the amount of noise but also increases the amount of computing necessary to obtain results. This section discusses the tradeoffs between the number of data records averaged and the uncertainty in the resulting cepstra where reverberation must be discerned and separated.

As a result of this additive noise, Eq.(2.9.4) is modified to become

$$y(t) = h_r(t) * \sum_i h_{pi}(t) * a_i \delta(t - \tau_i) + n(t) \quad (2.14.1)$$

or in the frequency domain

$$Y(\omega) = H_r(\omega) \sum_i a_i e^{-j\omega\tau_i} H_{pi}(\omega) + N(\omega) \quad (2.14.2)$$

If the average of the noise, $\overline{N(\omega)}$, is zero, we still have

$$\overline{Y(\omega)} = H_r(\omega) \sum_i \overline{a_i e^{-j\omega\tau_i}} \cdot \overline{H_{pi}(\omega)} \quad (2.14.3)$$

However, any estimate for $\overline{Y(\omega)}$, say $\hat{\overline{Y(\omega)}}$, possesses some fluctuation $\Delta[\overline{Y(\omega)}]$ about its mean. If for this analysis here we assume $\overline{H_{pi}(\omega)} = 1$, or that the average pulse shape is an ideal function, then

$$\begin{aligned} \hat{\overline{Y(\omega)}} &= \overline{Y(\omega)} + \Delta[\hat{\overline{Y(\omega)}}] = H_r(\omega) \left\{ \sum_i \overline{a_i e^{-j\omega\tau_i}} + \Delta \left[\sum_i \overline{a_i e^{-j\omega\tau_i}} \right] \right\} \\ &\quad + 0 + \Delta[\hat{\overline{N(\omega)}}] \end{aligned} \quad (2.14.4)$$

or

$$\begin{aligned} \log \{ \hat{\overline{Y(\omega)}} + \Delta[\hat{\overline{Y(\omega)}}] - \Delta[\hat{\overline{N(\omega)}}] \} &= \log H_r(\omega) \\ &\quad + \log \left\{ \sum_i \overline{a_i e^{-j\omega\tau_i}} + \Delta \left[\sum_i \overline{a_i e^{-j\omega\tau_i}} \right] \right\} \end{aligned} \quad (2.14.5)$$

The weak noise case will be considered here. Assuming all fluctuations are small compared to their averages as is in the case of high signal to noise ratios, then Eq. (2.14.5) becomes to a first approximation

$$\log \overline{\hat{Y}(\omega)} = \log \overline{Y(\omega)} + \Delta[\log \overline{\hat{Y}(\omega)}] \quad (2.14.6)$$

$$= \log \{ \overline{Y(\omega)} + \Delta[\overline{\hat{Y}(\omega)}] \} \quad (2.14.7)$$

$$\approx \log \overline{Y(\omega)} + \frac{\Delta[\overline{\hat{Y}(\omega)}]}{\overline{Y(\omega)}} \quad (2.14.8)$$

$$\approx \log H_r(\omega) + \log \left\{ \sum_i \overline{a_i e^{-j\omega\tau_i}} + \frac{\Delta[\overline{N(\omega)}]}{\overline{Y(\omega)}} + \frac{\Delta[\sum_i \overline{a_i e^{-j\omega\tau_i}}]}{\sum_i \overline{a_i e^{-j\omega\tau_i}}} \right\} \quad (2.14.9)$$

The last two terms are error terms showing the departure of $\log \overline{\hat{Y}(\omega)}$ from the corresponding noise-free answer. Terms of the form $\Delta[Y] \cdot \Delta[N]$ have been ignored and $\overline{Y(\omega)}$ is assumed not to be zero. Now separating out the fluctuations we find

$$\Delta[\log \overline{\hat{Y}(\omega)}] = \frac{\Delta[\overline{N(\omega)}]}{\overline{Y(\omega)}} + \frac{\Delta[\sum_i \overline{a_i e^{-j\omega\tau_i}}]}{\sum_i \overline{a_i e^{-j\omega\tau_i}}} \quad (2.14.10)$$

provided the estimator $\overline{Y(\omega)}$ is unbiased (i.e., $\overline{\overline{Y(\omega)}} = \overline{Y(\omega)}$). Various statistics of the fluctuations of $Y(\tau)$ can be computed using this approach.

Relaxing the restriction that $\overline{H_{pi}(\omega)} = 1$ will only increase the error slightly. The term affected is the last one in equation 2.14.10. A similar analysis as the foregoing one can be performed on the two dimensional dereverberation derivation which results in similar bounds on the fluctuations of the average estimates.

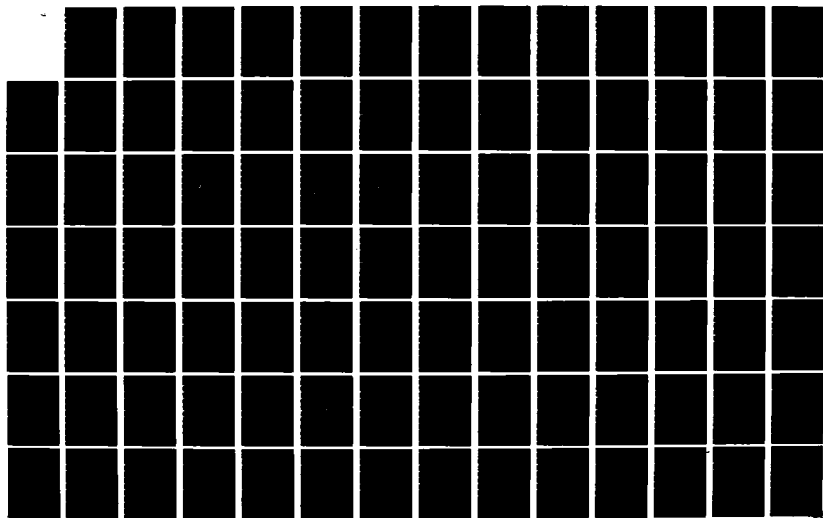
AD-A138 142

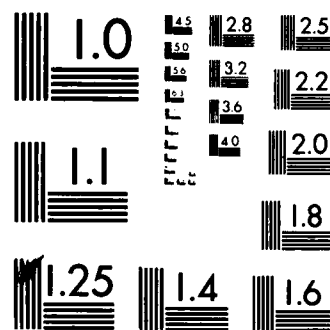
EMPLOYMENT OF ADAPTIVE LEARNING TECHNIQUES FOR THE
DISCRIMINATION OF ACOU. (U) GENERAL ELECTRIC CORPORATE
RESEARCH AND DEVELOPMENT SCHENECTA. J W ERKES ET AL.
NOV 83 83-SRD-060 N00014-82-C-2031 F/G 20/1

2/4

UNCLASSIFIED

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Turning now to the problem of noise added to the phase of the Fourier transform of the processed signal, this subsection will demonstrate how phase noise affects the phase unwrapping routine necessary for the computation of the cepstrum. As explained in [7] the phase unwrapping algorithm occasionally runs into ambiguities (of factors of 2π) in calculating the arguments. In such instances an intermediate point is evaluated between the last point and the new point where the ambiguity arises. This midpoint is used to resolve the ambiguity by its nearness to one of the two possibilities and hopefully direct the phase unwrapping routine to the correct possibility of the ambiguity. A problem arises if phase noise is present on the intermediate point that is large enough to place that point closer to the incorrect possibility of the ambiguity. This, in turn, will induce a discontinuity in the unwrapped phase of 2π and cause an erroneous cepstrum to be computed.

To obtain an upper bound on the probability of an incorrect phase and cepstrum given an amount of random phase noise due to measurement and numerical roundoff error, a derivation will be performed which indicates the amount of averaging on the ensemble records and the necessary length T of each record to achieve that bound. If $x(t)$ and $y(t)$ are two stationary random processes representing the lumped phase noise, it has been shown [82] that the mean values of $R_{xy}(\tau, T)$ over the ensemble of records of length T is

$$\langle R_{xy}(\tau, T) \rangle = \frac{1}{T} \int_0^T \langle x(t)y(t + \tau) \rangle dt \quad (2.14.11)$$

$$= \frac{1}{T} \int_0^T R_{xy}(\tau) dt \quad (2.14.12)$$

$$\langle R_{xy}(\tau, T) \rangle = R_{xy}(\tau) \quad (2.14.13)$$

The variance is defined by

$$\sigma_{xy}^2(\tau, T) = \langle R_{xy}^2(\tau, T) \rangle - R_{xy}^2(\tau) \quad (2.14.14)$$

where

$$\langle R_{xy}^2(\tau, T) \rangle = \frac{1}{T^2} \int_0^T \int_0^T \langle x(t)y(t+\tau)x(u)y(u+\tau) \rangle dt du \quad (2.14.15)$$

using the stationary hypothesis and the definition

$$P_{xy}^2(\tau, t, u) = \langle x(t)y(t+\tau)x(u)y(u+\tau) \rangle \quad (2.14.16)$$

then

$$P_{xy}^2(\tau, t, u) = P_{xy}^2(\tau, v) \quad ; \quad v = u - t \quad (2.14.17)$$

and substituting (2.14.17) into (2.14.15) with $dv = -dt$ gives

$$\langle R_{xy}^2(\tau, T) \rangle = \frac{1}{T^2} \int_0^T \int_{u-T}^u P_{xy}^2(\tau, v) dv du \quad (2.14.18)$$

Following the derivation in [82] it can be shown

$$\sigma_{xy}^2(\tau, T) = \frac{2}{T^2} \int_0^T (T-v) [P_{xy}^2(\tau, v) - R_{xy}^2(\tau)] dv \quad (2.14.19)$$

In this application the length of the records for averaging determines $\sigma_{xy}(\tau, T)$, the standard deviation on the probability of the intermediate phase point position. This standard deviation determines the probability that the intermediate phase point will be wrong and is equal to $Q(\kappa)$ where $Q(\cdot)$ is the area under the Gaussian tail and $\kappa = R_{xy}(\tau)/\sigma_{xy}(\tau, T)$. $Q(\kappa)$ is the probability of an erroneous jog and is equal to the probability of a cepstral error. $\sigma_{xy}(\tau, T)$ also puts a limit on the deviation of every point on the unwrapped phase from its true value and can be used to limit the phase noise for cases where no erroneous jogs are present from improperly selected ambiguities.

PART 3

DUAL SOURCE DEREVERBERATION

3.1 The Dual Source Problem Formulation

If two sources are present in the medium, each having the same form of time function as the single source discussed previously, then the signal $Y(\omega)$, measured at a sensor would be described by

$$Y(\omega) = H_{r1}(\omega) \left[\sum_i a_{i1} e^{-j\omega\tau_{i1}} \right] + H_{r2}(\omega) \left[\sum_k a_{k2} e^{-j\omega\tau_{k2}} \right] \quad (3.1.1)$$

Proceeding as before by averaging both sides of (3.1.1) gives

$$\overline{Y(\omega)} = \overline{H_{r1}(\omega) \left[\sum_i a_{i1} e^{-j\omega\tau_{i1}} \right] + H_{r2}(\omega) \left[\sum_k a_{k2} e^{-j\omega\tau_{k2}} \right]} \quad (3.1.2)$$

$$= \overline{H_{r1}(\omega) \left[\sum_i a_{i1} e^{-j\omega\tau_{i1}} \right]} + \overline{H_{r2}(\omega) \left[\sum_k a_{k2} e^{-j\omega\tau_{k2}} \right]} \quad (3.1.3)$$

$$= H_{r1}(\omega) Q_1(\omega) \overline{\sum_i a_{i1}} + H_{r2}(\omega) Q_2(\omega) \overline{\sum_k a_{k2}} \quad (3.1.4)$$

assuming the a_i 's and τ_i 's are independent.

Equation (3.1.4) first of all does not lead to a solution of $H_{r1}(\omega)$ or $H_{r2}(\omega)$ since there are two unknowns in one equation. Secondly it shows how the presence of an unknown source will destroy the information available on $H_{r1}(\omega)$.

Before proceeding on to another approach, some preliminary background mathematics will be covered for an array of sensors in a medium. For N sensors shown in Figure 3.1.1,

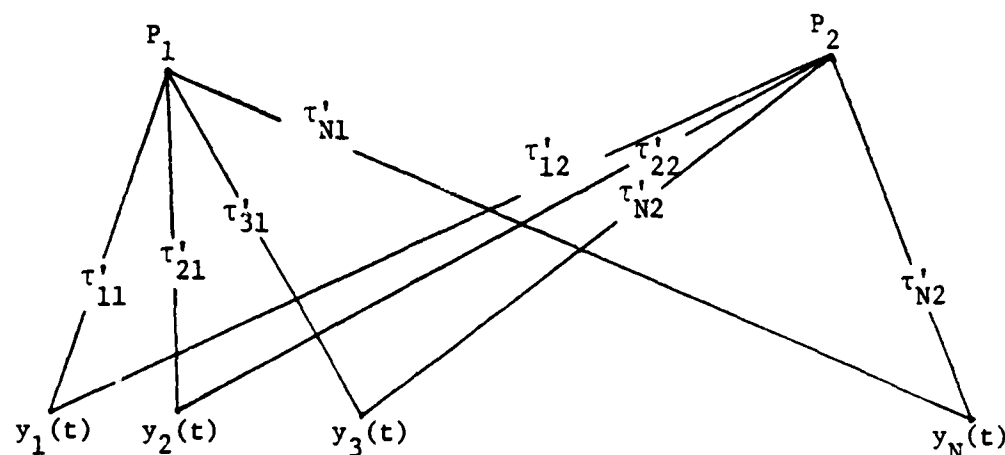


Figure 3.1.1. Signal Path Delays from Points P_1 and P_2 in a Medium to N Sensor Signals $y_i(t)$

the signal at sensor i from the source $v_1(t)$ at P_1 is

$$y_i(t) = v_1(t - \tau'_{i1}) \quad (3.1.5)$$

where τ'_{i1} is the path time delay from P_1 to the i^{th} element. The array pattern for the array "looking" at P_2 is

$$z(t) = \sum_{i=1}^N a_i y_i(t + \tau'_{i2}) \quad (3.1.6)$$

where a_i is the gain of sensor i . Substituting (3.1.5) into (3.1.6) gives

$$z(t) = \sum_{i=1}^N a_i v_1(t - \tau'_{i1} + \tau'_{i2}) = \int_{-\infty}^{\infty} V_1(\omega) \left[\sum_{i=1}^N a_i e^{-j\omega\tau_{i1}} \right] e^{j\omega t} d\omega \quad (3.1.7)$$

where $\tau_i = \tau'_{i1} - \tau'_{i2}$. Forming the averaged product

$$\overline{zz^*} = \frac{1}{T} \int \sum_{i=1}^N a_i v_1(t - \tau_i) \sum_{k=1}^N a_k^* v_1^*(t - \tau_k) dt \quad (3.1.8)$$

$$= \frac{1}{T} \int \sum_{i=1}^N \sum_{k=1}^N a_i a_k^* v_1(t - \tau_i) v_1^*(t - \tau_k) dt \quad (3.1.9)$$

$$= \sum_{i=1}^N \sum_{k=1}^N a_i a_k^* \frac{1}{T} \int v_1(t - \tau_i) v_1^*(t - \tau_k) dt \quad (3.1.10)$$

$$= \sum_{i=1}^N \sum_{k=1}^N a_i a_k^* R_{v_1 v_1}(\tau_i - \tau_k) = \sum_{i=1}^N \sum_{k=1}^N a_i a_k^* R_{y_i y_k}(\tau'_{i2} - \tau'_{k2}) \quad (3.1.11)$$

where

$$R_{y_i y_k}(\tau) = E[y_i(t) y_k(t - \tau)] = \frac{1}{T} \int y_i(t) y_k(t - \tau) dt \quad (3.1.12)$$

$$R_{v_1 v_1}(\tau) = E[v_1(t) v_1(t - \tau)] = \frac{1}{T} \int v_1(t) v_1(t - \tau) dt \quad (3.1.13)$$

is the beam equation for the array looking at a single point.

Defining

$$S_{v_1 v_1}(\tau) = \int_{-\infty}^{\infty} R_{v_1 v_1}(\tau) e^{-j\omega\tau} d\tau \quad (3.1.14)$$

then $\overline{zz^*}$ can be rewritten as

$$\overline{zz^*} = F^{-1} \left[F \left[\sum_{i=1}^N \sum_{k=1}^N a_i a_k^* R_{v_1 v_1}(\tau_i - \tau_k) \right] \right] \quad (3.1.15)$$

$$= \int_{-\infty}^{\infty} \sum_{i=1}^N \sum_{k=1}^N a_i a_k^* S_{v_1 v_1}(\omega) e^{j\omega(\tau_i - \tau_k)} d\omega \quad (3.1.16)$$

$$= \int_{-\infty}^{\infty} S_{v_1 v_1}(\omega) \sum_{i=1}^N \sum_{k=1}^N a_i a_k^* e^{j\omega\tau_i} e^{-j\omega\tau_k} d\omega \quad (3.1.17)$$

$$= \int_{-\infty}^{\infty} S_{v_1 v_1}(\omega) \sum_{i=1}^N a_i e^{j\omega\tau_i} \sum_{k=1}^N a_k^* e^{-j\omega\tau_k} d\omega \quad (3.1.18)$$

$$= \int_{-\infty}^{\infty} S_{v_1 v_1}(\omega) \left| \sum_{i=1}^N a_i e^{-j\omega\tau_i} \right|^2 d\omega \quad (3.1.19)$$

This last equation will be useful for describing the averaged array signal for the steered array.

Returning to the dual noise source in a reverberating channel, this can be shown graphically as in Figure 3.1.2.

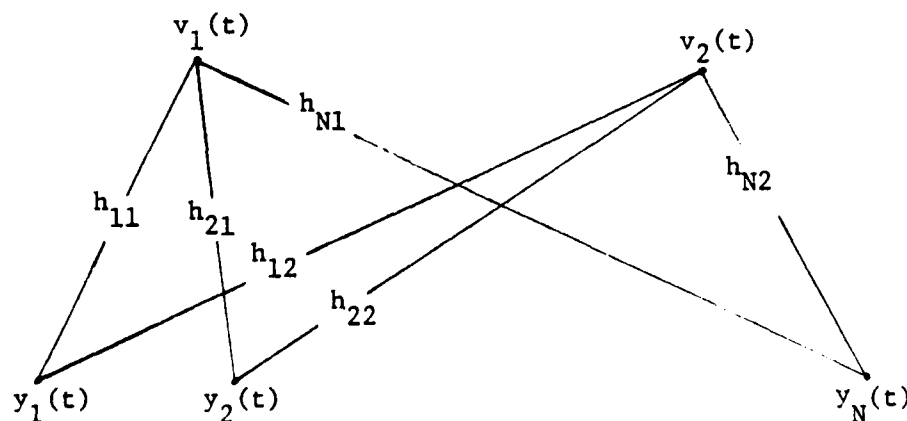


Figure 3.1.2 Path Impulse Functions from Sources $v_1(t)$ and $v_2(t)$ to N Sensors Signals $y_1(t)$

Here the h 's represent the path impulse functions and the v 's the noise sources. With $x_{ij}(t)$ defined as the signal from source j to sensor i convolved with the path transfer function as

$$x_{i1}(t) = v_1(t) * h_{i1}(t) \quad (3.1.20)$$

$$x_{i2}(t) = v_2(t) * h_{i2}(t) \quad (3.1.21)$$

and $y_i(t)$ the signal into sensor i , then

$$y_i(t) = a_i [x_{i1}(t) + x_{i2}(t)] \quad (3.1.22)$$

Now z becomes

$$z(t) = \sum_{i=1}^N a_i y_i(t - \tau_i) = \sum_{i=1}^N a_i [x_{i1}(t - \tau_{i1}) + x_{i2}(t - \tau_{i2})] \quad (3.1.23)$$

or

$$z(t) = \int_{-\infty}^{\infty} \{V_1(\omega) \left[\sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} \right] + V_2(\omega) \left[\sum_{k=1}^N H_{k2}(\omega) a_k e^{-j\omega\tau_{k2}} \right]\} e^{j\omega t} d\omega \quad (3.1.24)$$

and

$$\begin{aligned} \overline{zz^*} = & \int_{-\infty}^{\infty} \{S_{v_1 v_1}(\omega) \left| \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} \right|^2 + S_{v_2 v_2}(\omega) \left| \sum_{i=1}^N H_{i2}(\omega) a_i e^{-j\omega\tau_{i2}} \right|^2 \\ & + 2S_{v_1 v_2}(\omega) \left| \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} \right| \left| \sum_{i=1}^N H_{i2}(\omega) a_i e^{-j\omega\tau_{i2}} \right| \} d\omega \end{aligned} \quad (3.1.25)$$

If v_1 and v_2 are uncorrelated, then

$$\overline{zz^*} = \int_{-\infty}^{\infty} \{ S_{v_1 v_1}(\omega) \left| \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} \right|^2 + S_{v_2 v_2}(\omega) \left| \sum_{i=1}^N H_{i2}(\omega) a_i e^{-j\omega\tau_{i2}} \right|^2 \} d\omega \quad (3.1.26)$$

where

$$S_{v_i v_k}(\omega) = \int_{-\infty}^{\infty} E[v_i(t), v_k(t - \tau)] e^{-j\omega\tau} d\tau \quad (3.1.27)$$

Unfortunately this leads to one equation with $2N$ unknowns which can't be solved for the N H_{i1} 's and N H_{i2} 's.

However, taking the signal at the k^{th} sensor

$$y_k(t) = a_k [h_{k1}(t) * v_1(t) + h_{k2}(t) * v_2(t)] \quad (3.1.28)$$

$$= \int_{-\infty}^{\infty} a_k \{ H_{k1}(\omega) V_1(\omega) + H_{k2}(\omega) V_2(\omega) \} e^{j\omega t} d\omega \quad (3.1.29)$$

and forming the averaged product $\overline{zy_k^*}$

$$\begin{aligned} \overline{zy_k^*} &= \frac{1}{T} \int_T \{ \sum_{i=1}^N a_i [h_{i1}(t) * v_1(t - \tau_{i1}) \\ &\quad + h_{i2}(t) * v_2(t - \tau_{i2})] \} \{ a_k^* [h_{k1}(t) * v_1^*(t) \\ &\quad + h_{k2}(t) * v_2^*(t)] \} dt \end{aligned} \quad (3.1.30)$$

$$\begin{aligned}
\overline{zy_k^*} = & \sum_{i=1}^N \left\{ \frac{1}{T} \int_T a_i a_k^* [h_{i1}(t) * v_1(t - \tau_{i1}) \cdot h_{k1}(t) * v_1^*(t)] dt \right. \\
& + \frac{1}{T} \int_T a_i a_k^* [h_{i1}(t) * v_1(t - \tau_{i1}) \cdot h_{k2}(t) * v_2^*(t)] dt \\
& + \frac{1}{T} \int_T a_i a_k^* [h_{i2}(t) * v_2(t - \tau_{i2}) \cdot h_{k1}(t) * v_1^*(t)] dt \\
& \left. + \frac{1}{T} \int_T a_i a_k^* [h_{i2}(t) * v_2(t - \tau_{i2}) \cdot h_{k2}(t) * v_2^*(t)] dt \right\} \quad (3.1.31)
\end{aligned}$$

with

$$\begin{aligned}
& \frac{1}{T} \int_T a_i a_k [h_{i1}(t) * v_1^*(t - \tau_{i1}) \cdot h_{k1}(t) * v_1^*(t)] dt \\
& = a_i a_k^* h_{i1}(-\tau_{i1}) * h_{k1}(-\tau_{i1}) * R_{v_1 v_1}(-\tau_{i1}) \quad (3.1.32)
\end{aligned}$$

substituted into (3.1.31) yields

$$\begin{aligned}
\overline{zy_k^*} = & \sum_{i=1}^N a_i a_k^* \{ h_{i1}(-\tau_{i1}) * h_{k1}(-\tau_{i1}) * R_{v_1 v_1}(-\tau_{i1}) \\
& + h_{i1}(-\tau_{i1}) * h_{k2}(-\tau_{i1}) * R_{v_1 v_2}(-\tau_{i1}) \\
& + h_{i2}(-\tau_{i2}) * h_{k1}(-\tau_{i2}) * R_{v_2 v_1}(-\tau_{i2}) \\
& + h_{i2}(-\tau_{i2}) * h_{k2}(-\tau_{i2}) * R_{v_2 v_2}(-\tau_{i2}) \} \quad (3.1.33)
\end{aligned}$$

or in terms of the Fourier transform

$$\begin{aligned}
\overline{zy_k^*} = & \int_{-\infty}^{\infty} \sum_{i=1}^N a_i a_k^* \{ H_{i1}(\omega) H_{k1}(\omega) S_{v_1 v_1}(\omega) e^{-j\omega\tau_{i1}} \\
& + H_{i1}(\omega) H_{k2}(\omega) S_{v_1 v_2}(\omega) e^{-j\omega\tau_{i1}} + H_{i2}(\omega) H_{k1}(\omega) S_{v_2 v_1}(\omega) e^{-j\omega\tau_{i2}} \\
& + H_{i2}(\omega) H_{k2}(\omega) S_{v_2 v_2}(\omega) e^{-j\omega\tau_{i2}} \} d\omega
\end{aligned} \tag{3.1.34}$$

Regrouping terms produces

$$\begin{aligned}
\overline{zy_k^*} = & \int_{-\infty}^{\infty} \{ a_k S_{v_1 v_1}(\omega) H_{k1}(\omega) \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} \\
& + a_k^* S_{v_1 v_2}(\omega) H_{k2}(\omega) \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} \\
& + a_k^* S_{v_2 v_1}(\omega) H_{k1}(\omega) \sum_{i=1}^N H_{i2}(\omega) a_i e^{-j\omega\tau_{i2}} \\
& + a_k^* S_{v_2 v_2}(\omega) H_{k2}(\omega) \sum_{i=1}^N H_{i2}(\omega) a_i e^{-j\omega\tau_{i2}} \} d\omega
\end{aligned} \tag{3.1.35}$$

If v_1 and v_2 are independent then $S_{v_2 v_1} = S_{v_1 v_2} = 0$ and

$$\begin{aligned}
\overline{zy_k^*} = & \int_{-\infty}^{\infty} \{ a_k^* S_{v_1 v_1}(\omega) H_{k1}(\omega) \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} \\
& + a_k^* S_{v_2 v_2}(\omega) H_{k2}(\omega) \sum_{i=1}^N H_{i2}(\omega) a_i e^{-j\omega\tau_{i2}} \} d\omega
\end{aligned} \tag{3.1.36}$$

With the array looking at v_1 , the second term in (3.1.36) will be suppressed and

$$\overline{zy_k^*} = \int_{-\infty}^{\infty} \{a_k^* S_{v_1 v_1}(\omega) H_{k1}(\omega) \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega \tau_{i1}}\} d\omega \quad (3.1.37)$$

The error in (3.1.37) will not be negligible if the array has a significant side lobe "looking" at v_2 or if the two sources are slightly dependent. For this situation, two error terms must be included in the expression and

$$\overline{zy_k^*} = \int_{-\infty}^{\infty} a_k^* S_{v_1 v_1}(\omega) H_{k1}(\omega) \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega \tau_{i1}} d\omega + \epsilon_{v_2 v_2} + \epsilon_{v_1 v_2} \quad (3.1.38)$$

Numerical evaluation of the terms in (3.3.37) may be achieved by delaying y_k to form

$$\begin{aligned} \overline{z(t)y_k^*(t - \tau)} &= \int_{-\infty}^{\infty} a_k^* S_{v_1 v_1}(\omega) H_{k1}(\omega) \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega(\tau_{i1} - \tau)} d\omega \\ &+ \int_{-\infty}^{\infty} a_k^* S_{v_2 v_2}(\omega) H_{k2}(\omega) \sum_{i=1}^N H_{i2}(\omega) a_i e^{-j\omega(\tau_{i2} - \tau)} d\omega \end{aligned} \quad (3.1.39)$$

$$\approx \int_{-\infty}^{\infty} a_k^* S_{v_1 v_1}(\omega) H_{k1}(\omega) \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega(\tau_{i1} - \tau)} d\omega \quad (3.1.40)$$

Taking the Fourier transform of both sides of (3.1.40) with respect to τ yields

$$K_k(\omega) = a_k S_{v_1 v_1}(\omega) H_{k1}(\omega) \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} \quad (3.1.41)$$

where

$$K_k(\omega) = \int_{-\infty}^{\infty} z(t) y_k^*(t-\tau) e^{-j\omega\tau} dt \quad (3.1.42)$$

Letting $S = S_{v_1 v_1}(\omega)$, $H_i = H_{i1}(\omega)$, and $a_i = a_i e^{-j\omega\tau_{i1}}$, (3.1.41) can be rewritten as

$$K_k = S H_k [H_1 a_1 + H_2 a_2 + \dots + H_N a_N] ; k=1,2, \dots N \quad (3.1.43)$$

This is a set of N nonlinear quadratic equations to be solved at each frequency, ω , of interest. The N equations come from the N possible cross-products of z and y_i , $i=1,2, \dots N$. The solution to this set of equations follows in the next section.

For the special case of impulsive signals, $S_{v_1 v_1}(\omega) = Q(\omega)Q(\omega)^* \overline{a_i a_j}$ and $S_{v_1 v_1}(\omega)$ can be removed from (3.1.41) using the cepstral techniques described in Section 2.4. Thus

$$K_k(\tau) = H_{k1}(\tau) + \int_{-\infty}^{\infty} \log[H_{11}(\omega) a_1 e^{-j\omega\tau_{11}} + \dots + H_{N1}(\omega) a_N e^{-j\omega\tau_{N1}}] e^{j\omega\tau} d\omega \quad (3.1.44)$$

for $\tau \neq 0$ and a_i independent of τ_i . Inverting the cepstrum leaves

$$K_k(\omega) \approx H_{k1}(\omega) [H_{11}(\omega) a_1 e^{-j\omega\tau_{11}} + \dots + H_{N1}(\omega) a_N e^{-j\omega\tau_{N1}}] \quad (3.1.45)$$

3.2 Solution to the Dual Source Nonlinear Quadratic Equations

The N nonlinear quadratic equations are as follows:

$$\begin{aligned}
 K_1 &= H_1[a_1H_1 + a_2H_2 + \dots + a_NH_N] \\
 K_2 &= H_2[a_1H_1 + a_2H_2 + \dots + a_NH_N] \\
 &\vdots \\
 K_N &= H_N[a_1H_1 + a_2H_2 + \dots + a_NH_N]
 \end{aligned} \tag{3.2.1}$$

Multiplying both sides of equation i by a_i for $i = 1, 2, \dots, N$ gives

$$\begin{aligned}
 a_1K_1 &= a_1H_1[a_1H_1 + a_2H_2 + \dots + a_NH_N] \\
 a_2K_2 &= a_2H_2[a_1H_1 + a_2H_2 + \dots + a_NH_N] \\
 &\vdots \\
 a_NK_N &= a_NH_N[a_1H_1 + a_2H_2 + \dots + a_NH_N]
 \end{aligned} \tag{3.2.2}$$

Now adding all N equations together yields

$$\begin{aligned}
 a_1K_1 + a_2K_2 + \dots + a_NK_N \\
 = [a_1H_1 + a_2H_2 + \dots + a_NH_N][a_1H_1 + a_2H_2 + \dots + a_NH_N]
 \end{aligned} \tag{3.2.3}$$

and

$$\sqrt{a_1K_1 + a_2K_2 + \dots + a_NK_N} = [a_1H_1 + a_2H_2 + \dots + a_NH_N] = C \tag{3.2.4}$$

Substituting back into the N equations in (3.2.1) results in

$$K_k = H_k[a_1H_1 + a_2H_2 + \dots + a_NH_N] \tag{3.2.5}$$

which implies

$$H_k = \frac{K_k}{C} : k=1,2, \dots N \quad (3.2.6)$$

This is a closed form solution of the N nonlinear quadratic equations to be calculated for each frequency, ω , of interest.

3.3 Improving the Accuracy of H_{i1} and H_{i2} and Image Enhancement

The incomplete suppression of V_2 by the array "looking" at V_1 will give rise to errors in the calculations of the $N H_{i1}(\omega)$ terms and likewise when steered to V_2 for the $N H_{i2}(\omega)$ terms. These errors are due to the imperfect beam pattern of the array and are largest when, while the array is "looking" at V_1 , V_2 is directly in the path of one of the array's largest side lobes. Obviously one way of reducing these errors would be to optimize the array pattern to reduce side and grating lobes to negligible amplitudes. The second and more expensive way with regard to computation time is to reduce the errors by iterating on the solution, using each previously calculated estimate of the H's to improve the next approximation.

The theory behind this procedure is that with two sources in the medium, the small error in measuring the H_{i1} 's is a known function of the H_{i2} 's and vice versa. The first estimate of the H_{i1} 's is plugged into this known function to reduce the error in the H_{i2} approximations. Now with an approximation for the H_{i2} 's, the H_{i1} values can be recalculated with a smaller error. This process can be

iterated until the changes in H_{i1} and H_{i2} from one iteration to the next is negligibly small.

The algorithm for this procedure follows.

- 1) Set $L_{k1} = L_{k2} = 0$.
- 2) Aim the array at V_1 and solve for H_{k1} with $K_k = \overline{zy_k^*} - L_{k2}$; $k = 1, 2, \dots, N$ using (3.2.4) and (3.2.6) (for all ω).
- 3) Calculate $L_{k1} = H_{k1}(\omega) \left[\sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} \right]$; $k = 1, 2, \dots, N$
- 4) Aim the array at V_2 and solve for H_{k2} with $K_k = \overline{zy_k^*} - L_{k1}$; $k = 1, 2, \dots, N$ using (3.2.4) and (3.2.6) (for all ω).
- 5) Calculate $L_{k2} = H_{k2}(\omega) \left[\sum_{i=1}^N H_{i2}(\omega) a_i e^{-j\omega\tau_{i2}} \right]$; $k = 1, 2, \dots, N$.
- 6) Go back to (2) unless the changes in the H_{i1} 's and H_{i2} 's are sufficiently small.

Image enhancement by the removal of the echos of a medium containing two sources must be done in two steps. The echos due to each source must be removed separately. The necessary analysis will be given for a medium containing two independent sources. The image of the field is $z_{xy}(t)$ and is obtained by calculating $z(t)$ of Eq. (3.1.23) at every point on some cartesian grid spanning the field. It is assumed that the grid is fine enough to permit easy resolution of the peak and null points of the field.

In order to deresonate the signal $z_{xy}(t)$, the peaks in the image must be separated and classified as to which of the two sources they are caused by. To achieve this, it is desirable to find

$z_0(t)$ at a peak which is due to only one source where $z_0(t)$ is $z(t)$ evaluated at a particular peak. This will insure that all the peaks in the time averaged product $\overline{z_{xy}z_0(t)}$ will be only the peaks in the total image $\overline{z_{xy}(t)}$ that are due to the source which formed the peak $z_0(t)$. Finding a suitable $z_0(t)$ can best be done by trial and error. First choose a large peak in $\overline{z_{xy}(t)}$ and set $z_0(t)$ equal to that value. Form the product $\overline{z_{xy}z_0(t)}$ and compare it to $\overline{z_{xy}^2(t)}$. If it has fewer peaks than $\overline{z_{xy}^2(t)}$, then $z_0(t)$ is uniquely formed by one of the two sources and is a good peak. If not, the field will have to be searched for another $z_0(t)$. Another peak, $z'_0(t)$ must also be found which is uniquely formed by the second source. Again trial and error is the best method for finding a solution. The check for a good $z'_0(t)$ is that the product $\overline{z_{xy}z'_0(t)}$ has substantially different peaks and nulls than $\overline{z_{xy}z_0(t)}$ and that the sum of $\overline{z_{xy}z_0(t)}$ and $\overline{z_{xy}z'_0(t)}$ is similar to $\overline{z_{xy}^2(t)}$.

Deresonation is performed by simply multiplying the signals from the N sensors, $Y_i(\omega)$, $i = 1, 2, \dots, N$ by the inverse of the path transfer function from one of the sources to the sensor. These are the inverses of the H_{i1} and H_{i2} calculated at the beginning of this section. Assuming $z_0(t)$ is due to source 1 and $z'_0(t)$ to source 2, then the deresonated source 1 image is calculated from $z_{c1}(t)$ where

$$z_{c1}(t) = \int_{-\infty}^{\infty} \left\{ \sum_{i=1}^N a_i Y_i(\omega) H_{i1}^{-1}(\omega) e^{-j\omega\tau_i} \right\} e^{j\omega t} d\omega \quad (3.3.1)$$

by calculating $\overline{z_{c1}z_0(t)}$. Again by adjusting the τ_i 's, this value

can be found at every point on the cartesian grid spanning the field. Likewise the deresonated image of source 2 is found from $\overline{z_{c2}z_0'(t)}$ where

$$z_{c2}(t) = \int_{-\infty}^{\infty} \left\{ \sum_{i=1}^N a_i Y_i(\omega) H_{i2}^{-1}(\omega) e^{-j\omega\tau_{i1}} \right\} e^{j\omega t} d\omega \quad (3.3.2)$$

The total deresonated field is the sum of the two deresonated parts, $\overline{z_{c1}z_0(t)}$ and $\overline{z_{c2}z_0'(t)}$. Ideally this sum would contain only two peaks; one at each of the positions of the two noise sources. Thus the positions of the sources could unambiguously be determined.

3.4 Convergence of Iterative Solution of H_{i1} and H_{i2}

The solution of the nonlinear quadratic equations derived in Section 3.2 will be slightly in error due to the presence of the nonzero error terms shown in (3.1.38). Ignoring the $\epsilon_{v_1 v_2}$ term for now and concentrating on $\epsilon_{v_2 v_2}$ it will be demonstrated how the iteration procedure outlined in Section 3.3 will always converge on the exact answer and reduce the error after each iteration rather than diverge.

The basic equations being solved are

$$\begin{aligned} K_{k1} = & H_{k1} [H_{11} a_1 e^{-j\omega\tau_{11}} + \dots + H_{N1} a_N e^{-j\omega\tau_{N1}}] + H_{k2} [H_{12} a_1 e^{-j\omega\tau_{12}} \\ & + \dots + H_{N2} a_N e^{-j\omega\tau_{N2}}] ; k = 1, 2, \dots, N \end{aligned} \quad (3.4.1)$$

$$K_{k2} = H_{k1} [H_{11}a_1 e^{+j\omega\tau_{11}} + \dots + H_{N1}a_N e^{+j\omega\tau_{N1}}] + H_{k2} [H_{12}a_1 e^{+j\omega\tau_{12}} + \dots + H_{N2}a_N e^{+j\omega\tau_{N2}}] ; k = 1, 2, \dots, N \quad (3.4.2)$$

Letting

$$p = H_{11}a_1 e^{-j\omega\tau_{11}} + \dots + H_{N1}a_N e^{-j\omega\tau_{N1}} \quad (3.4.3)$$

$$q = H_{12}a_1 e^{-j\omega\tau_{12}} + \dots + H_{N2}a_N e^{-j\omega\tau_{N2}} \quad (3.4.4)$$

$$r = H_{11}a_1 e^{+j\omega\tau_{11}} + \dots + H_{N1}a_N e^{+j\omega\tau_{N1}} \quad (3.4.5)$$

$$s = H_{12}a_1 e^{+j\omega\tau_{12}} + \dots + H_{N2}a_N e^{+j\omega\tau_{N2}} \quad (3.4.6)$$

then the problem is of the form

$$K_1 = pH_1 + qH_2 \quad (3.4.7)$$

$$K_2 = rH_1 + sH_2 \quad (3.4.8)$$

where it is assumed $q < p$ and $r < s$ as is the case when the array is steered off the secondary source. The first iteration assumes qH_2 is zero so contains an error term ϵ .

$$K_1 = pH_1 + \epsilon \quad (3.4.9)$$

so

$$H_1 = \frac{K_1}{p} - \frac{1}{p} \epsilon \quad (3.4.10)$$

For the second source

$$K_2 = r \left[\frac{K_1 - \epsilon}{p} \right] + sH_2 \quad (3.4.11)$$

and so

$$H_2 = \frac{K_2}{s} - \frac{r}{ps} K_1 + \frac{r}{rs} \epsilon \quad (3.4.12)$$

After the second iteration

$$H_1 = \frac{1}{p} K_1 - \frac{q}{ps} K_2 + \frac{qr}{p^2 s} K_1 - \frac{qr}{p^2 p} \epsilon \quad (3.4.13)$$

and

$$H_2 = \frac{1}{s} K_2 - \frac{r}{ps} K_1 + \frac{qr}{p^2 s^2} K_2 - \frac{qr^2}{p^2 s^2} K_1 + \frac{qr}{p^2 s^2} \epsilon \quad (3.4.14)$$

In general, after the k^{th} iteration

$$H_1 = \frac{1}{p} K_1 + \dots - \frac{q^{k-1} r^{k-1}}{p^k s^{k-1}} \epsilon \quad (3.4.15)$$

and

$$H_2 = \frac{1}{s} K_2 + \dots + \frac{q^{k-1} r^{k-1}}{p^k s^k} \epsilon \quad (3.4.16)$$

Therefore it can be seen that the error in H_1 and H_2 approaches zero whenever the assumption $q < p$ and $r < s$ holds or less restrictively when $qr < ps$.

3.5 The Three Source Problem

When three sources rather than two are present in the medium, the equations are slightly different and the procedure used to arrive at a solution must be modified although the same basic technique is still used. Equation (3.1.24) becomes

$$\begin{aligned}
 z = \int_{-\infty}^{\infty} \{ & v_1(\omega) \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} + v_2(\omega) \sum_{i=1}^N H_{i2}(\omega) a_i e^{-j\omega\tau_{i2}} \\
 & + v_3(\omega) \sum_{i=1}^N H_{i3}(\omega) a_i e^{-j\omega\tau_{i3}} \} e^{-j\omega\tau} d\omega
 \end{aligned} \quad (3.5.1)$$

and if v_1 , v_2 and v_3 are uncorrelated, (3.1.36) becomes

$$\begin{aligned}
 \overline{zy_k^*} = \int_{-\infty}^{\infty} \{ & S_{v_1 v_1}(\omega) H_{k1}(\omega) \sum_{i=1}^N H_{i1}(\omega) a_i e^{-j\omega\tau_{i1}} \\
 & + S_{v_2 v_2}(\omega) H_{k2}(\omega) \sum_{i=1}^N H_{i2}(\omega) a_i e^{-j\omega\tau_{i2}} \\
 & + S_{v_3 v_3}(\omega) H_{k3}(\omega) \sum_{i=1}^N H_{i3}(\omega) a_i e^{-j\omega\tau_{i3}} \} d\omega
 \end{aligned} \quad (3.5.2)$$

Here again for an array steered to look at v_1 the last two terms in (3.5.2) will be small.

This fact can be used to improve the initial estimates of H_{i1} , H_{i2} and H_{i3} by iteration using the same technique as in Section 2.4. The algorithm is as follows:

- 1) Set $L_{k1} = L_{k2} = L_{k3} = 0$.
- 2) Aim the array at v_1 and solve for H_{k1} with

$$K_k = \overline{zy_k^*} - L_{k2} - L_{k3} \quad ; \quad k = 1, 2, \dots, N \text{ using (3.2.4) and (3.2.6)}$$
 (for all ω).
- 3) Calculate $L_{k1} = H_{k1}(\omega) \sum_{i=1}^N H_{i1} a_i e^{-j\omega\tau_{i1}} \quad ; \quad k = 1, 2, \dots, N$.

- 4) Aim the array at V_2 and solve for H_{k2} with

$$K_k = \overline{zy_k^*} - L_{k1} - L_{k3} \quad ; \quad k = 1, 2, \dots, N \text{ using (3.2.4) and (3.2.6) (for all } \omega).$$

- 5) Calculate $L_{k2} = H_{k2}(\omega) \sum_{i=1}^N H_{i2} a_i e^{-j\omega\tau_{i2}} \quad ; \quad k = 1, 2, \dots, N.$

- 6) Aim the array at V_3 and solve for H_{i3} with

$$K_k = \overline{zy_k^*} - L_{k1} - L_{k2} \quad ; \quad k = 1, 2, \dots, N \text{ using (3.24) and (3.26) (for all } \omega).$$

- 7) Calculate $L_{k3} = H_{k3}(\omega) \sum_{i=1}^N H_{i3} a_i e^{-j\omega\tau_{i3}} \quad ; \quad k = 1, 2, \dots, N.$

- 8) Calculate back to (2) unless H_{k1} , H_{k2} and H_{k3} are sufficiently accurate.

3.6 Array Dereverberation Simulations

In order to test out the theory previously derived for dual source dereverberation, a set of simulations was performed. Two independent acoustic sources were located in a medium in front of a reflective surface which would create a simple echo from each source. In this type of situation the echo can be treated as a virtual second source directly behind the reflective boundary and equidistant from it as the real source. Figure 3.6.1 shows the relative positions of the real sources in front of the boundary, the virtual echo sources behind the boundary and the positions of the four sensors used to image the medium. The four edges of the image plane in this simulation represent perfectly matched non-reflective edges. The source

pulses were taken to be two-sided decaying exponential acoustic emission as shown previously in Figure 2.12.7.

Simulations were done initially on simpler systems. In the first case shown in Figure 3.6.2 the right source was completely suppressed and the reflective surface was removed to show the correlation image of a single, non-reverberating signal. Figure 3.6.3 is a similar image of the right source alone without its echo. Both figures show large peaks where the sources are located correctly but with extra ripples in the fore- and background due to correlations of separate pairs of received signals. These would be greatly suppressed if a larger number of sensors had been simulated instead of the four used here to limit the excessive costs of the simulations with even this relatively few number of signals being correlated. This problem will show up on all the simulations in this section.

Figures 3.6.4 and 3.6.5 are again images of single sources alone, left and right respectively, however this time the reflective boundary is included which adds an echo to all the signals. The figures show how the echos form a peak almost as large as the source peak at the correct location but also create some other very large peaks around the source position. Already it would be difficult to locate the true source when there is only one emitting an acoustic signal. Single source dereverberation techniques derived in Part 2 produce images very similar to the ideal simulations of Figures 3.6.2 and 3.6.3, respectively.

When both sources are active simultaneously with the inclusion of the echos off the reflective boundary and the signals are processed using straight correlation techniques, the image produced is shown in Figure 3.6.6. Here is a classic example of the complete loss of the ability to locate the sources due to the reverberation and the existence of two sources. There are multiple high peaks distributed throughout the plane of observation and in some cases the maximum peaks are not even located over either a real source or virtual echo position.

As a benchmark image against which to test the dereverberation processed image, a simulation was done of the two sources operating together without echos. This image is shown in Figure 3.6.7 and represents an image formed after perfect and ideal dereverberation. Ideal dereverberation is in general impossible because an antiresonance which removes the echo of one source will cause an additional resonance on the second source and echo. In some cases, the deresonated signal may be worse for imaging than the original signal with echos. Figure 3.6.7 clearly shows the portion of the sources. Again the image would be improved if more sensors and signals were used for imaging by reducing the size of the ridges emanating from the position of the sensors.

Figure 3.6.8 shows the best result of the dual source dereverberation simulation of this test example. For this image it was determined that it was better not to deresonate the signals from one source that reached a given sensor later than the signal from the

source closer to the given sensor. As explained previously, this seemed to worsen the reconstruction. Figure 3.6.8 was constructed by deresonating the left half and right half separately and then joining the two halves together. Dereverberation loses amplitude information which explains why the left source appears smaller than the right. In general, this image is almost as good as the ideal dereverberated image with the most notable loss being in the relative amplitudes of the peaks marking the course locations to the remaining "noise" peaks and ridges. Overall, this simulation is very encouraging and shows a great deal of improvement over the unprocessed image of Figure 3.6.6. To help in comparing Figures 3.6.6 and 3.6.8, both have been reproduced together in Figure 3.6.9. It may take a few iterations of guessing at probable locations of sources in Figure 3.6.6 to obtain the deresonated image, but once it is found the sources can be pinpointed with accuracy and assurance of correctness.

As an example of an intermediate iteration for the deresonation process, Figure 3.6.10 shows a case where the location of the right source was correctly guessed but an incorrect guess was made for the location of the left source. In this case the location of the left source was taken mistakenly to be at the position of its echo which didn't lead to an obvious solution after processing.

Figures 3.6.11 to 3.6.14 show a typical convergence of the iterative solutions to eq. (3.4.7) and (3.4.8). Here H_1 is $H_{11}(\omega)$, the transfer function from the left source with its echo to the

left-most sensor, and H_2 is $H_{12}(\omega)$, the function of the right source to the same sensor. Figure 3.6.11 shows the two ideal channel responses obtained by activating each source along. Figure 3.6.12 represents the initial approximation to $H_{11}(\omega)$ and $H_{12}(\omega)$ by solving

$$H_{11}(\omega) = K_1(\omega)/p \quad ; \quad \text{all } \omega \quad (3.6.1)$$

and

$$H_{12}(\omega) = K_2(\omega)/s - rH_{11}(\omega)/s \quad ; \quad \text{all } \omega \quad (3.6.2)$$

Successive approximations to $H_{11}(\omega)$ and $H_{12}(\omega)$ are obtained by iteratively solving

$$H_{11}(\omega) = K_1(\omega)/p = qH_{21}(\omega)/p \quad ; \quad \text{all } \omega \quad (3.6.3)$$

and

$$H_{21}(\omega) = K_2(\omega)/s - rH_{11}(\omega)/s \quad ; \quad \text{all } \omega \quad (3.6.4)$$

Figure 3.6.13 shows the solution after 5 iterations and 3.6.14 is that after 20 iterations. At this point there was negligible change in the solution with further iterations and the solution was almost exactly equal to the ideal solution of Figure 3.6.11. For this example the ratio $|qr/ps|$ was slightly greater than .9 showing that for even relatively large ratios the convergence is very fast and accurate. $H_{21}(\omega)$ converges a little more rapidly and is close to its final value after 5 iterations. This proves the results of section 3.4 are valid in typical equations.

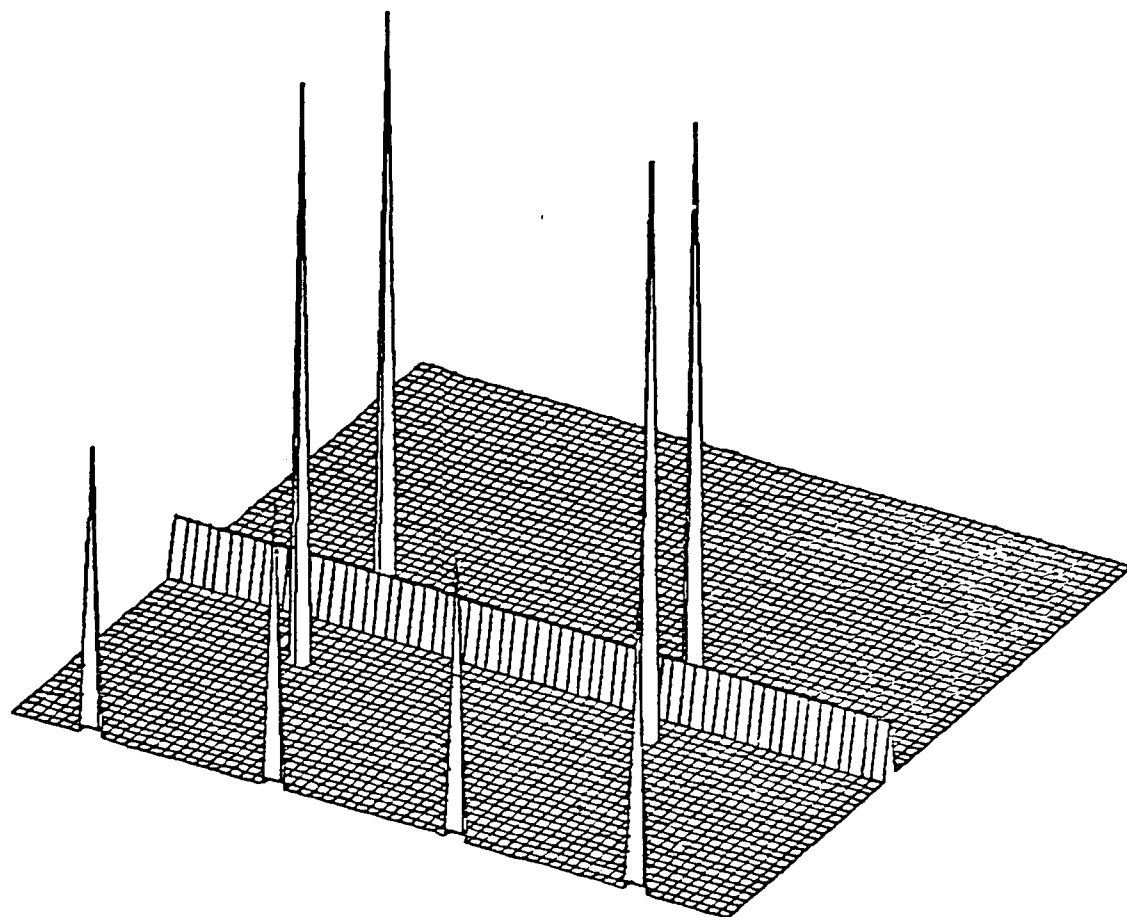


Figure 3.6.1 Plot of Position of Sources (Tall Peaks), Sensors (Short Peaks), and Reflective Boundary.

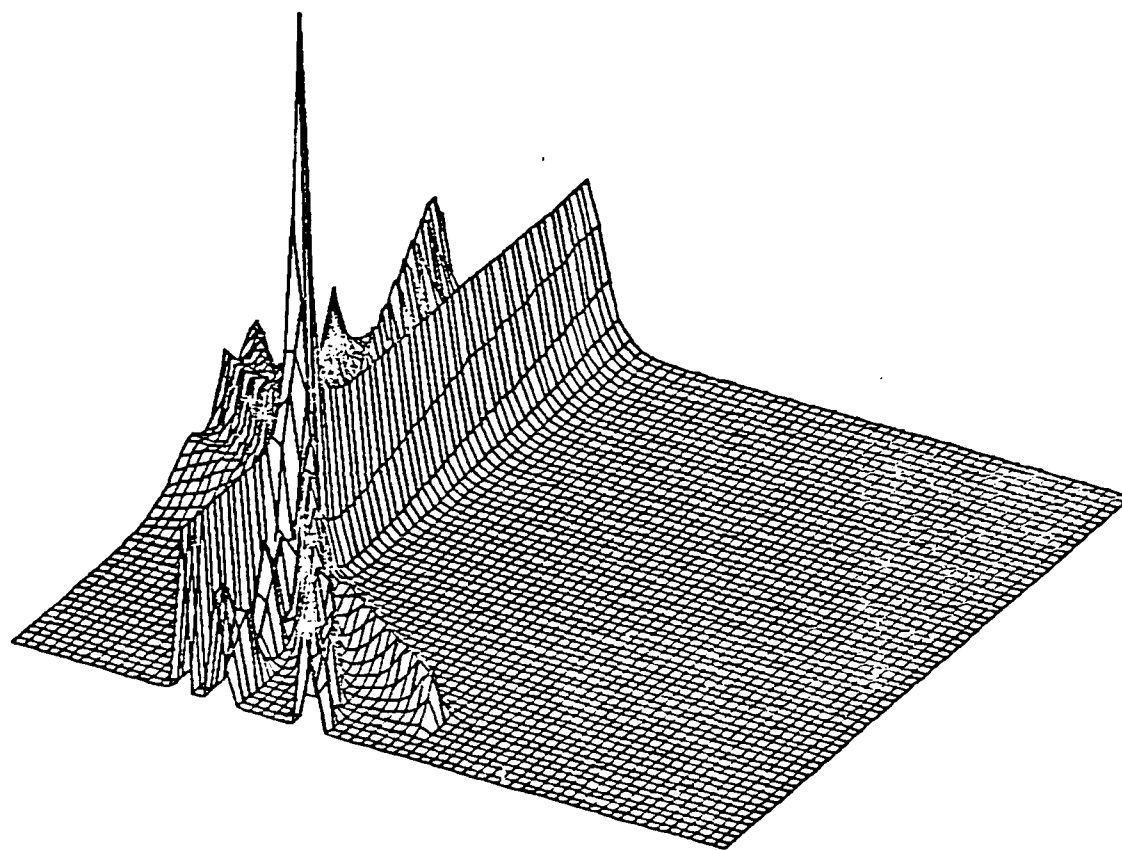


Figure 3.6.2 Image of Left-Source Activated Alone
Without Echo.

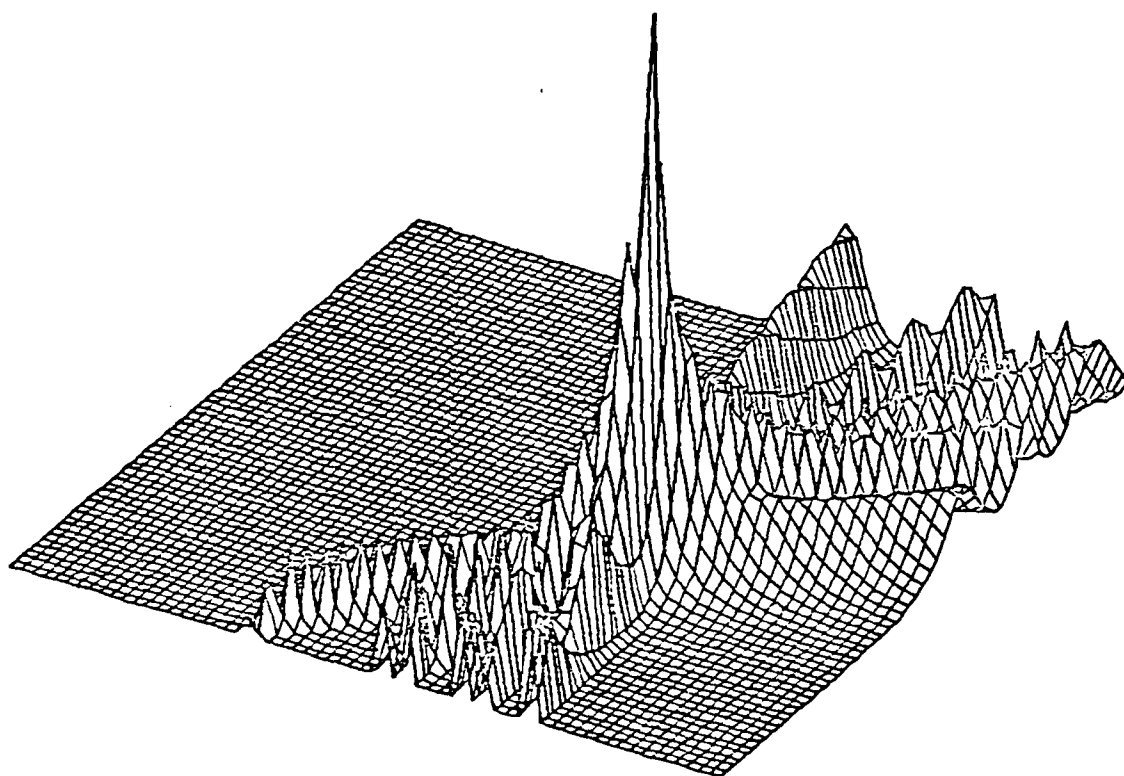


Figure 3.6.3 Image of Right Source Activated Alone Without Echo.

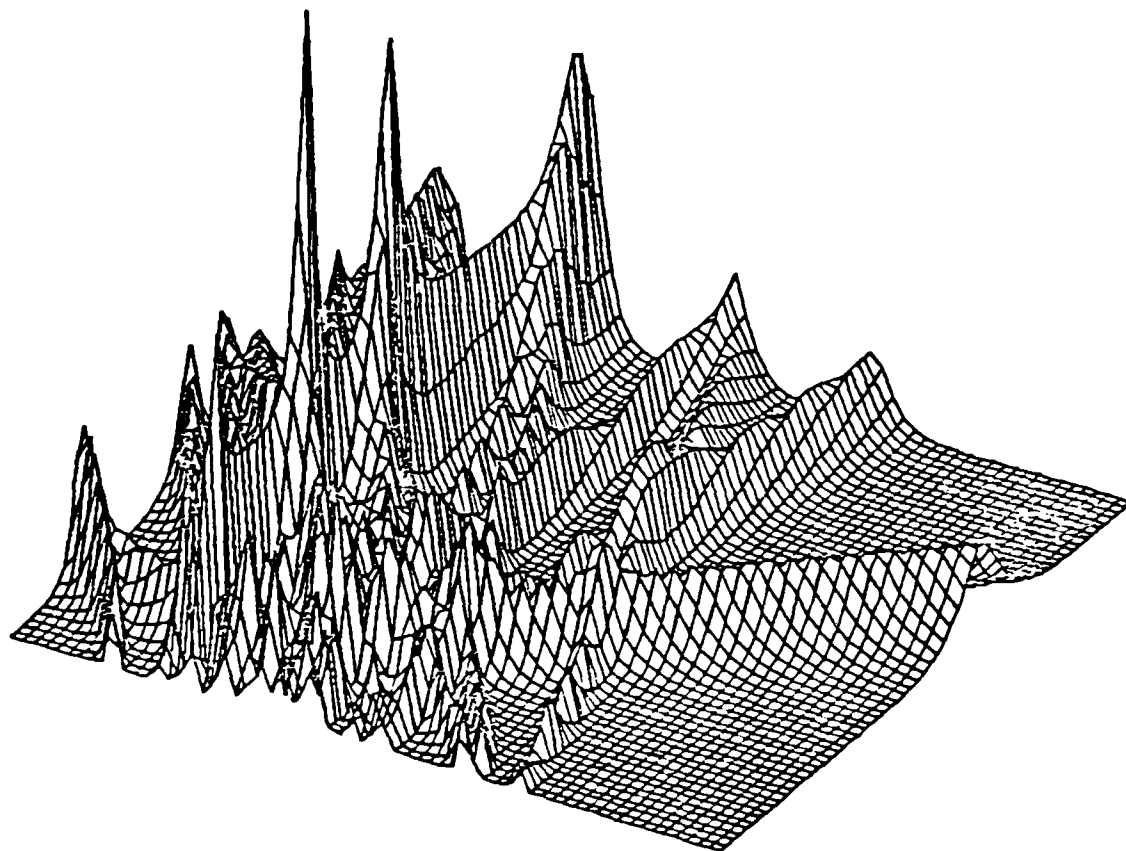


Figure 3.6.4 Image of Left Source Activated Alone
With Echo.

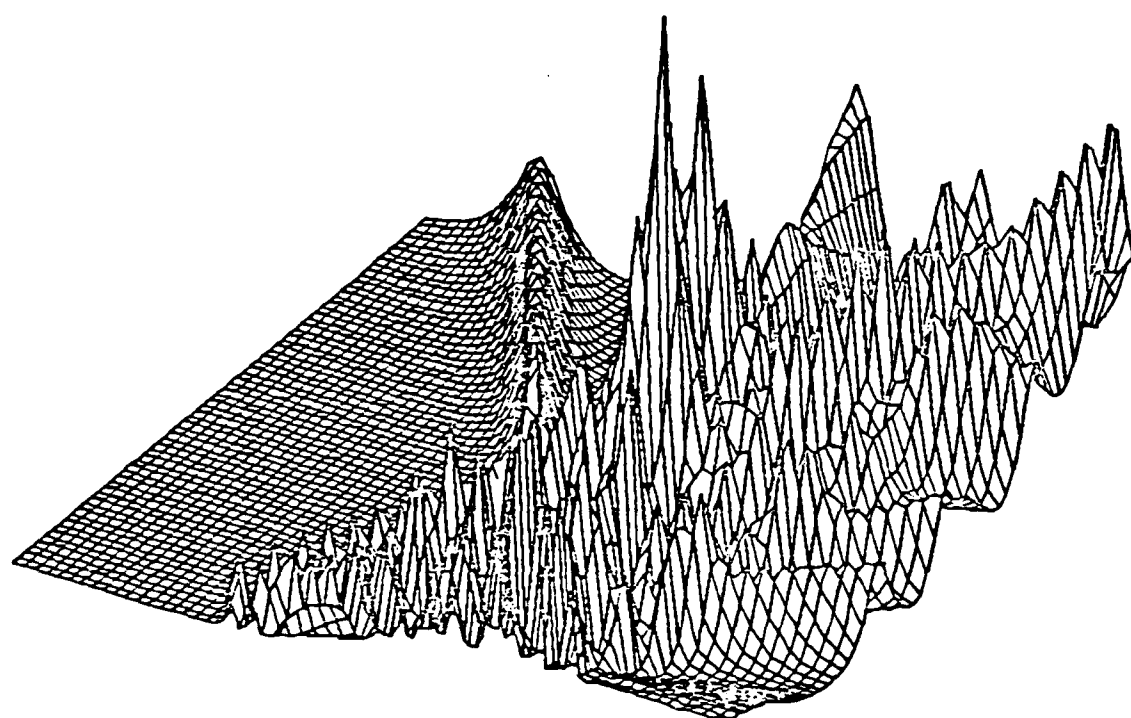


Figure 3.6.5 Image of Right Source Activated Alone
With Echo.

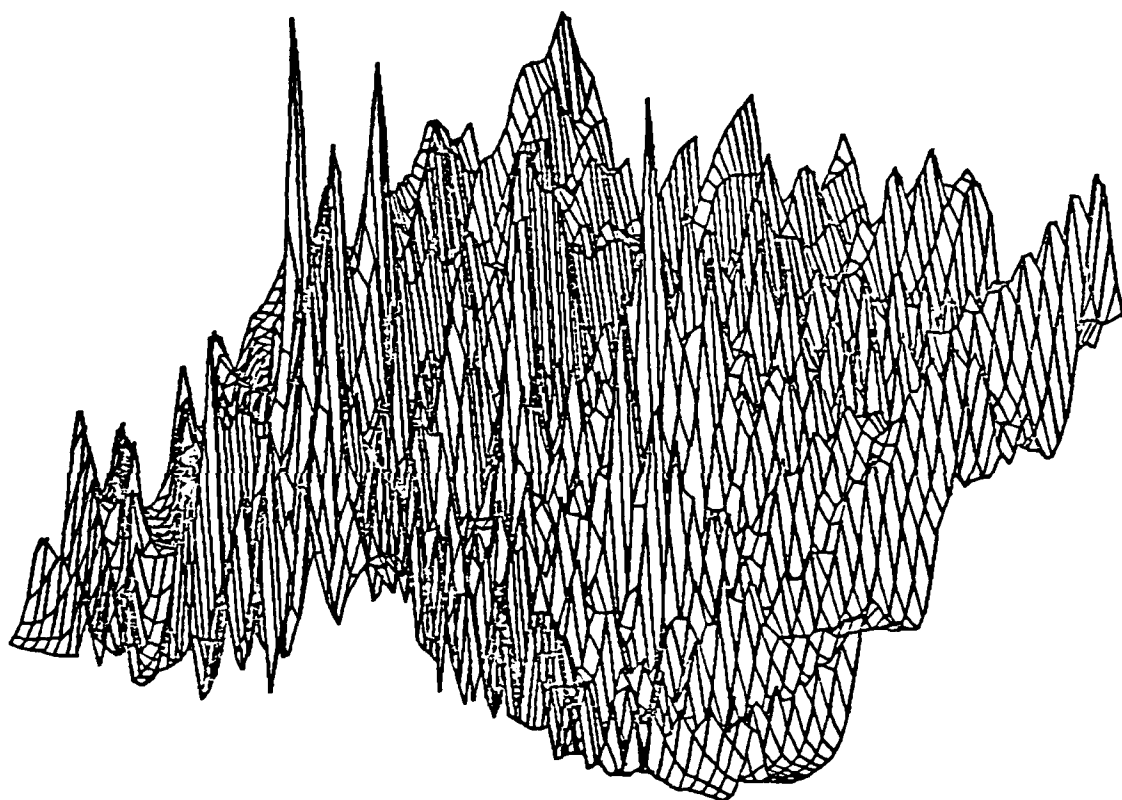


Figure 3.6.6 Image of Both Sources Activated Simultaneously With Echos.

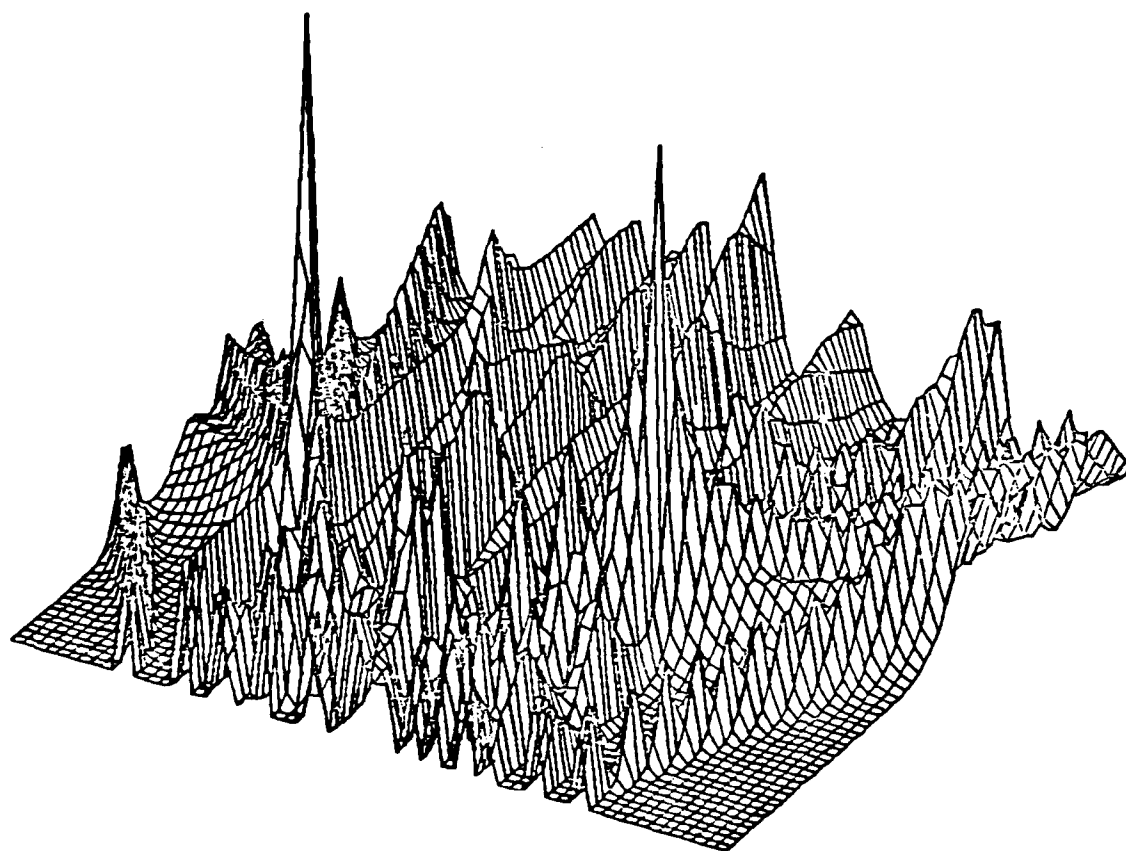


Figure 3.6.7 Image of Both Sources Activated Simultaneously Without Echos (Ideal Dereverberation Case).

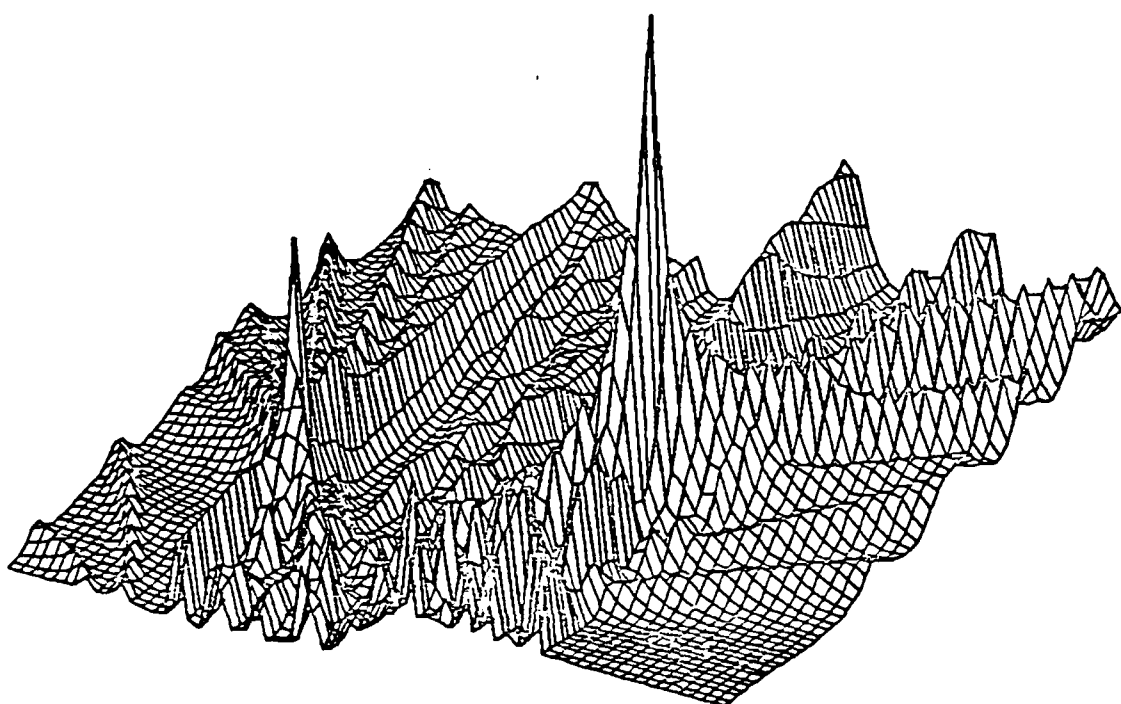


Figure 3.6.8 Dereverberated Image of Figure 3.6.6

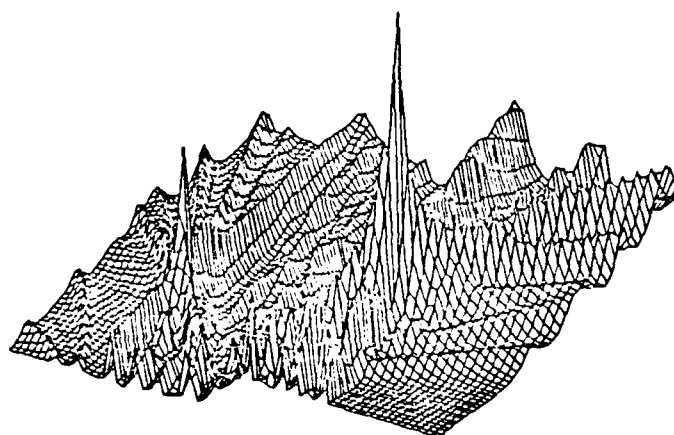
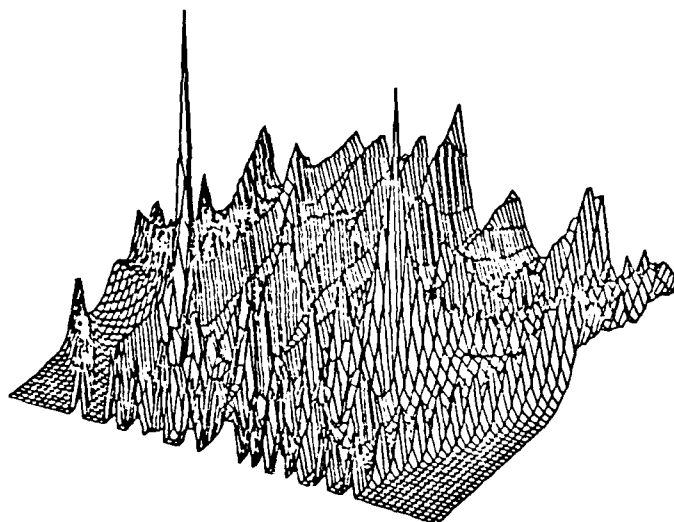


Figure 3.6.9 Comparison of Ideal Echoless Image (top) and Dereverberated Image (bottom).

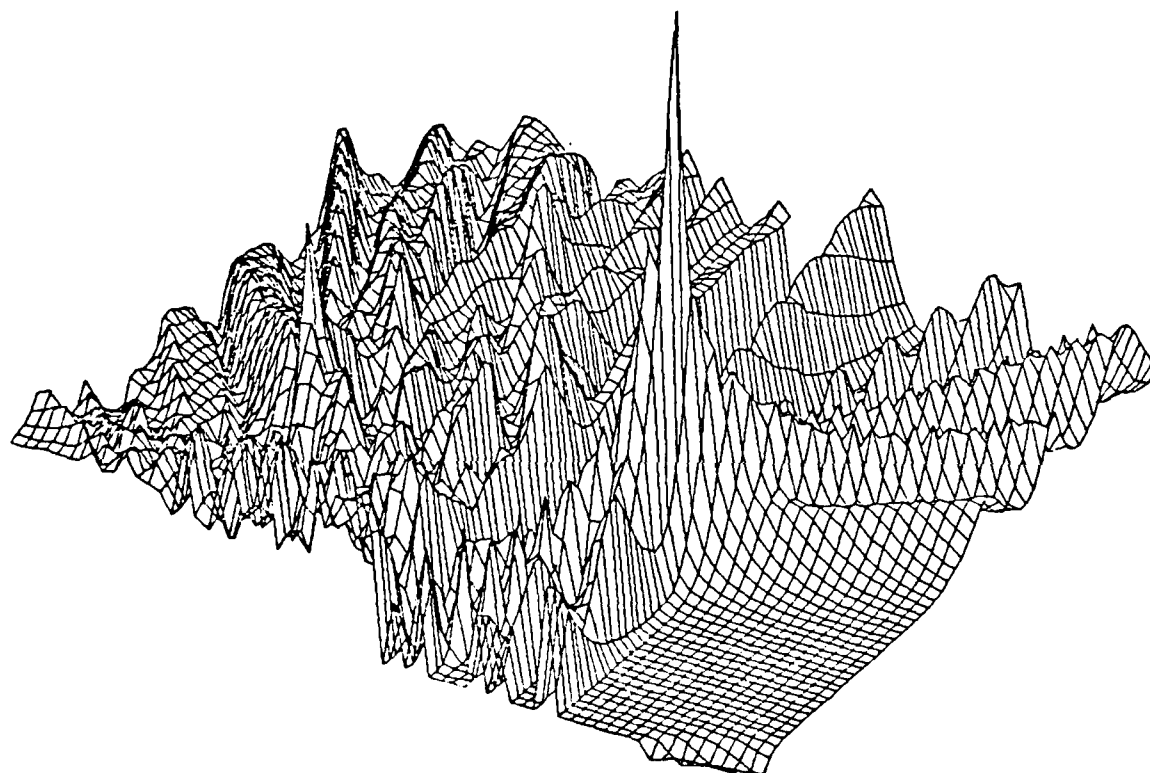


Figure 3.6.10 Intermediate Dereverberation Results of
Figure 3.6.8.

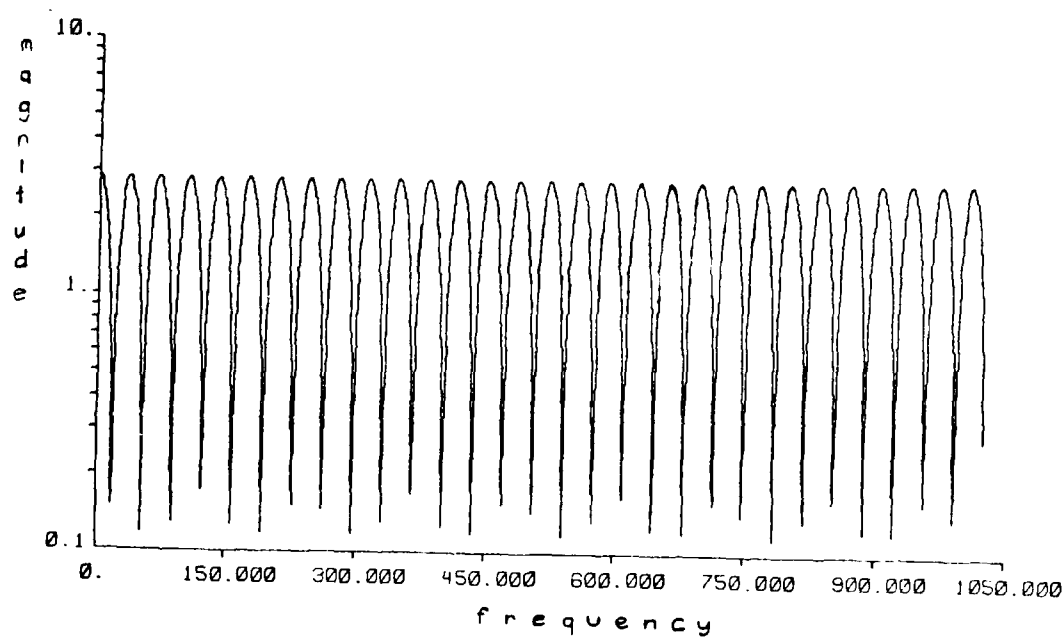
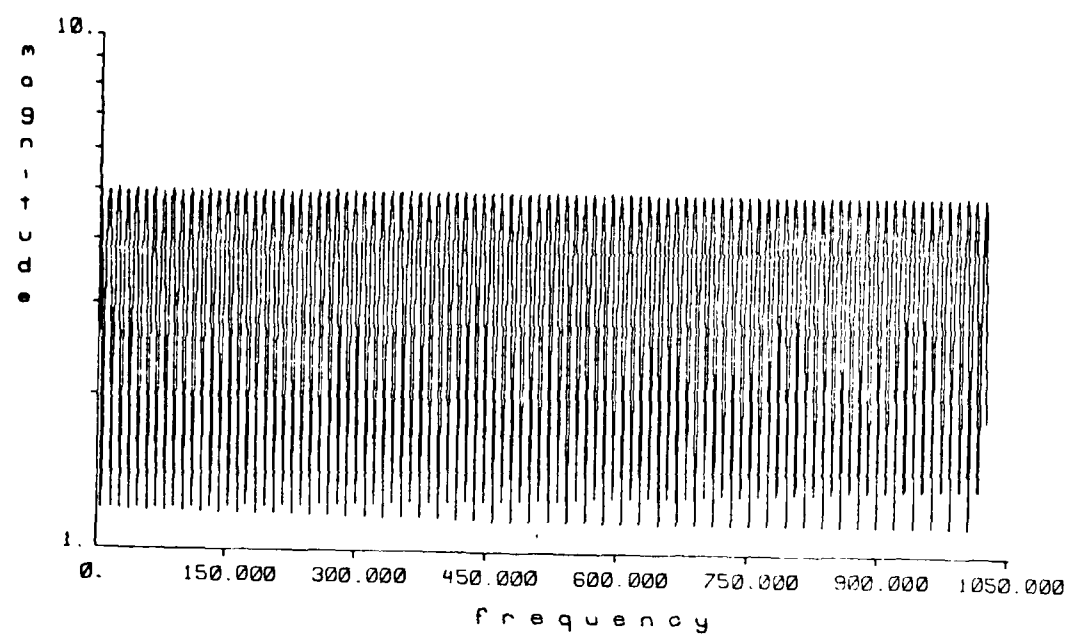


Figure 3.6.11 Ideal Channel Transfer Functions $H_{11}(\omega)$ (top) and $H_{21}(\omega)$ (bottom).

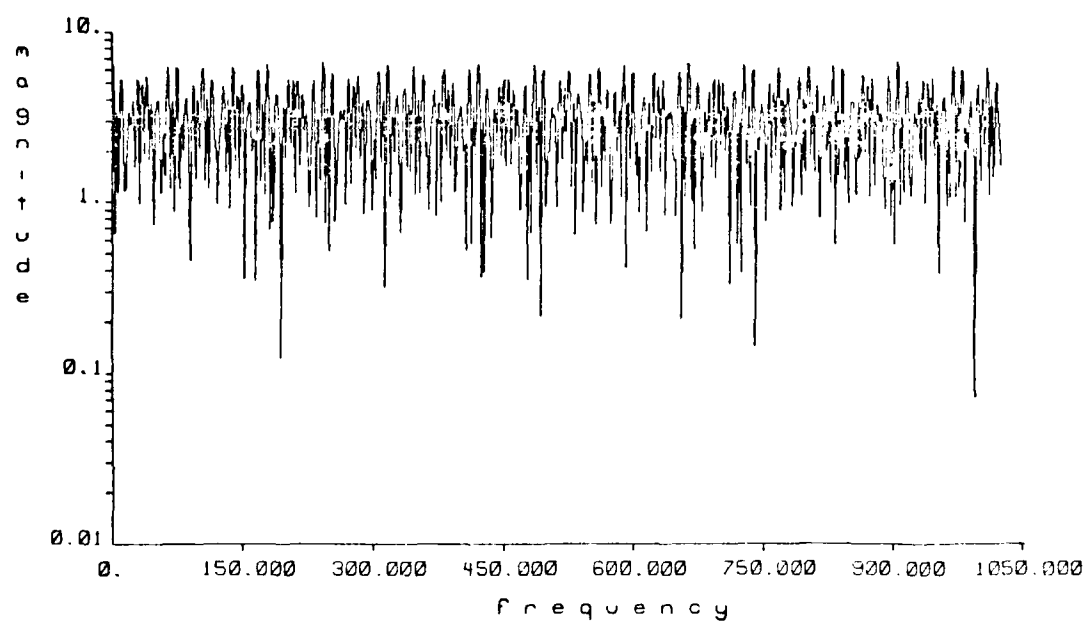
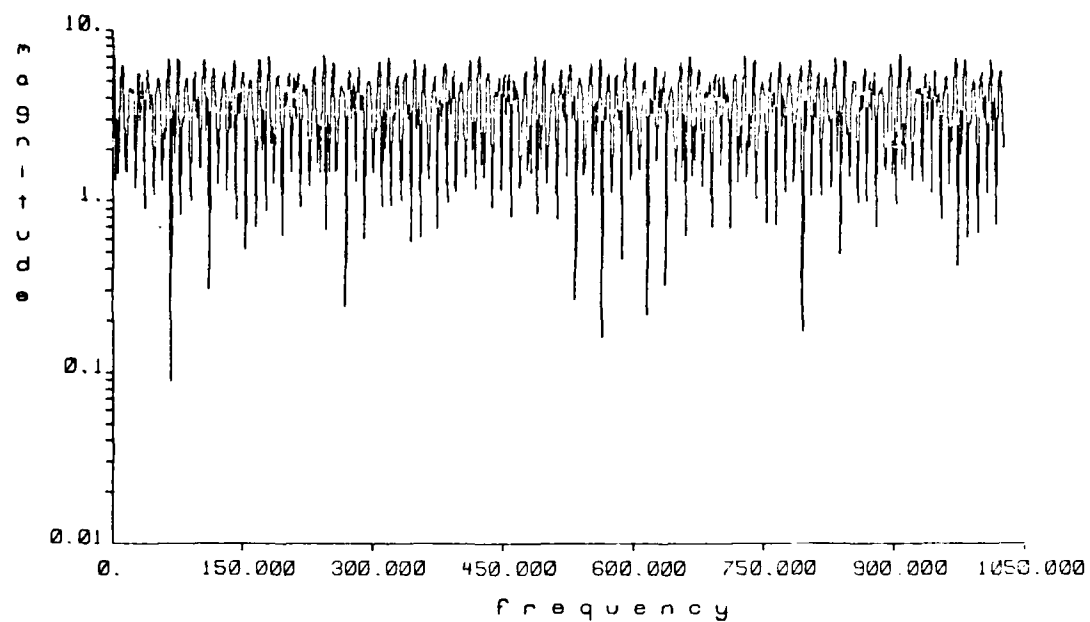


Figure 3.6.12 First Approximation of Channel Transfer Functions $H_{11}(\omega)$ (top) and $H_{21}(\omega)$ (bottom).

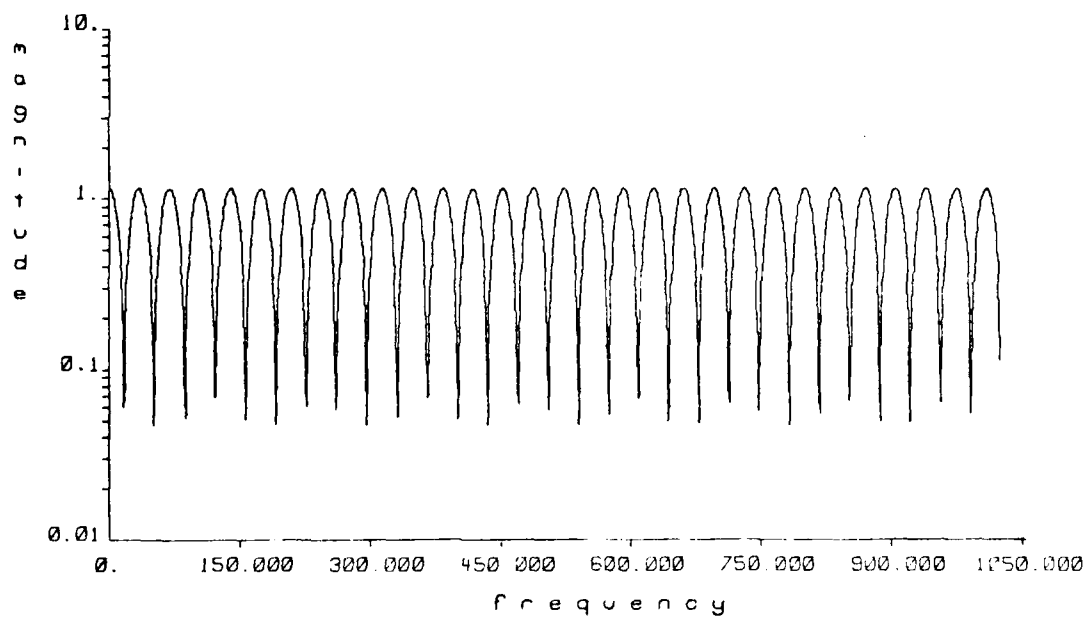
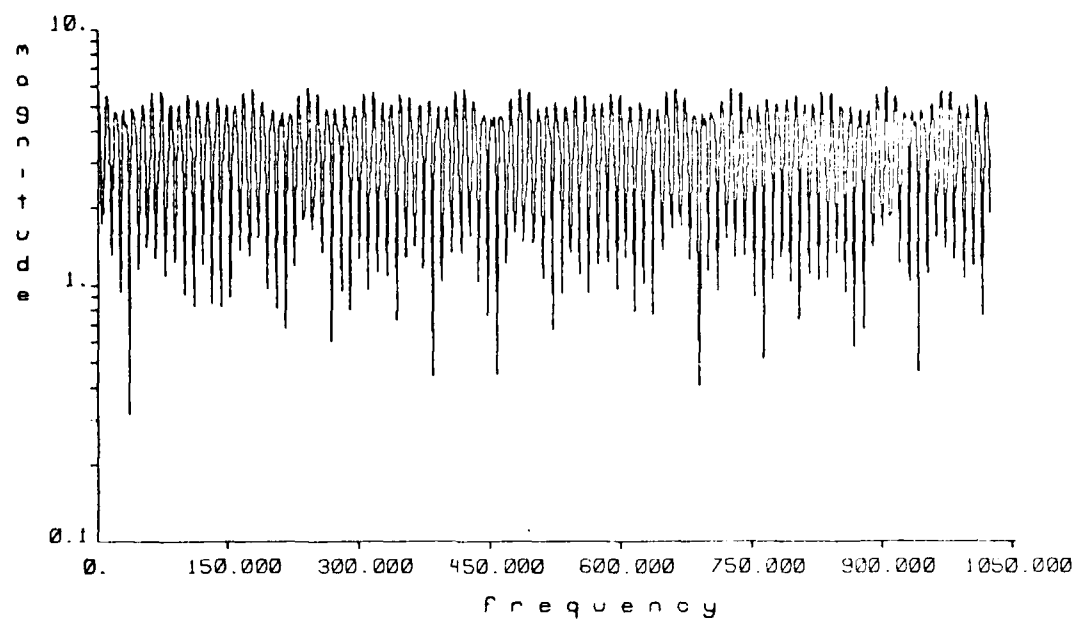


Figure 3.6.13 Approximation of Channel Transfer Functions $H_{11}(\omega)$ (top) and $H_{21}(\omega)$ (bottom) after 5 Iterations.

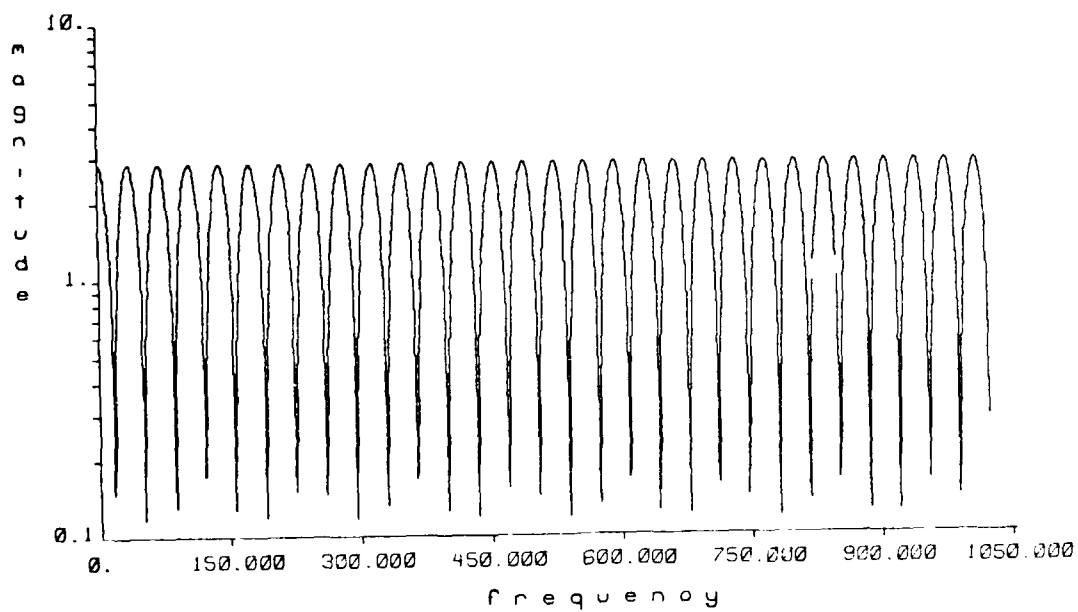
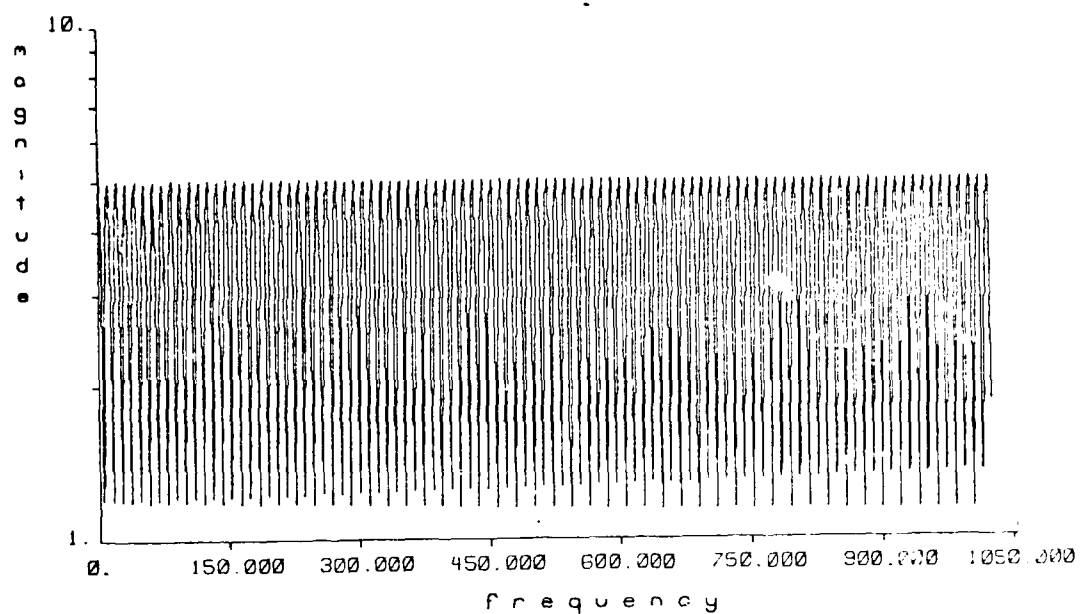


Figure 3.6.14 Approximation of Channel Transfer Functions $H_{11}(\omega)$ (top) and $H_{21}(\omega)$ (bottom) After 20 Iterations.

PART 4

ARRAY OPTIMIZATION

4.1 Phased Array Fundamentals

Recently there has been interest in applying the results of digital filter design theory to the optimization of the beam forming pattern of antenna arrays. The techniques employed usually involve minimax criteria for side lobe suppression. Certain applications require focused operation in order to make use of the near field for imaging. Nonuniform array element spacings have been successfully employed to reduce side lobes in phased arrays and to minimize the number of elements.

The Remez gain adjustment procedure has been shown to be effective in suppressing aperture side lobes but ineffective in controlling grating lobes for the far field [80]. This presents problems when the array is steered. The purpose of this chapter is to examine the robustness of the optimal far field gain optimization when the array is focused and steered in the Fresnel near field. The general conclusion is that the solutions for optimal gains are quite robust.

Presented here will be a combination of nonuniform spacing, focusing, and element shading using some optimizing window designs to form a narrow center beam with very good side lobe suppression. The treatment is unique in that it explicitly incorporates the effects of focusing in the optimization procedure.

Additionally, in designing multi-element phased arrays, whether for radar antennas or ultrasonic imaging systems, it is always desirable to keep the number of sensor elements in the array to a minimum. This results in economic savings in the construction of the array and in computational savings in the processing of the collected data for the reconstruction of images. With this reduction in the number of elements comes an immediate trade-off in the array's performance between image resolution and the introduction of ambiguities into the image through grating lobes. This chapter also presents results on the optimization of phased arrays for a specified number of elements with respect to grating lobes, side lobes and image resolution. This is accomplished by adjusting the spacings as well as the individual gains of the elements. The results are applied to one and two dimensional Cartesian arrays and also to Fresnel lens arrays.

The typical physical phased array system involves an array of transducers uniformly spaced in either one or two dimensions. Due to the proximity of objects to be imaged in various applications, the ultrasonic near field (which extends several meters from arrays of typical dimensions) must be properly characterized. Also, to improve imaging resolution it is necessary to focus the phased array. Discussed here will be special design problems that arise when phased arrays are to be operated in the near field and steered through wide viewing angles.

Analysis of phased arrays in the Fresnel region is essential since it is not possible to achieve significant focusing in the Fraunhofer limit of the ultrasonic field [48]. Unfortunately, conventional antenna theory design techniques are most effective for far field interference pattern optimization [49,50]. The purpose of this thesis will be to explore the limits of applicability for these conventional design techniques for near field operations and to suggest an approach to extending these techniques into the domain of interest in ultrasonic imaging.

Standard antenna operation involves steered but unfocused wave interference. The resolution achievable by such systems will generally not be acceptable for ultrasonic imaging as can be seen from the following argument. The angular spread of collimated radiation from an ideal unfocused aperture is approximately given by [51]

$$\theta \approx \lambda / (2D) \quad (4.1.1)$$

where λ is the wavelength of ultrasonic radiations and D is the diameter of the aperture. The distance, z_t , from the aperture to the point of transition between the near and far field is approximately [52]

$$z_t \approx D^2 / \lambda \quad (4.1.2)$$

Therefore, the width of the phased array main diffraction lobe will be roughly

$$w_t \approx \theta z_t = \left(\frac{\lambda}{2D}\right) \left(\frac{D^2}{\lambda}\right) = \frac{D}{2} \quad (4.1.3)$$

at that point. In other words the width of the unoptimized diffraction lobe at the point of emergence into the far field (where a clear main lobe can be seen) will be roughly equal to the size of the aperture, D , independent of wavelength. Of course, some applications may find adequate resolution attainable with small aperture diameters, but the angular collimation will usually be quite poor in this case as can be seen from equation (4.1.1), and the range of usefulness of such arrays is extremely limited.

On the other hand, the spot size s of a lens (simulating a focused array) is approximately given by [53]

$$s \approx \lambda \left(\frac{F}{D} \right) \quad (4.1.4)$$

where F is the lens focal length and D is now the lens aperture. If F were equal to the distance of the transition region, z_t , then

$$s \approx \lambda \left(\frac{z_t}{D} \right) = \frac{\lambda}{D} \left(\frac{D^2}{\lambda} \right) = D \quad (4.1.5)$$

so significant focusing is achievable only with F substantially less than z_t and preferably with a small F/D ratio. Values of F/D below unity usually place the focal point in unacceptable near field interference side lobes. Hence, useful focused unoptimized phased array operation will lie in the range of distances

$$D \leq F \leq z_t = D^2/\lambda \quad (4.1.6)$$

In this range the depth of focus, ℓ , can be estimated by

$$\ell = 4\lambda (F/D)^2 \quad (4.1.7)$$

$$= 4s (F/D) \quad (4.1.8)$$

so that again low F/D values are preferable if depth resolution is accomplished primarily by quasioptical methods. Since many ultrasonic phased arrays achieve depth information by pulsed operation, short depth of focus may not be a strong requirement.

The determining factor for requirement of short depth of focus is the bandwidth capability of the phased array which is a function of the technology used to achieve interelement isolation. Good interelement isolation is necessary in steered or focused arrays in order to impose the proper phase of excitation on the radiating elements. On-line compensation for interelement coupling is computationally very difficult and hence physical interelement isolation is preferable. These physical techniques frequently are effective only over some finite operational frequency bandwidth. This in turn has an effect on the depth resolution obtainable by pulsed operation. Some physical array designs achieve as high as 80% of available bandwidth but with loss of interelement isolation. More typically bandwidths are on the order of 30-50%.

Another problem in phased array imaging is caused by the inherent array grating lobes. In general the diffraction pattern for a phased array possesses both aperture and grating sidelobes. For a linear uniform array of aperture length D and element spacing d , the grating lobe angular spacing is roughly

$$\theta_g = \lambda/d \quad (4.1.9)$$

while the aperture lobes are roughly spaced in angle by

$$\theta_a \equiv \lambda/D \quad (4.1.10)$$

Rewriting Eq. 4.1.3, the main lobe width is related to D at a distance z by

$$w = \left(\frac{\lambda}{2D}\right) z \quad (4.1.11)$$

For good imaging purposes it is desirable to have w as small as possible and θ_g as large as possible. This insures high resolution with no ambiguities in the form of aliased image shadows. These aliases are due to grating lobes resulting from inadequate spatial sensor density. At this point we note that these are mutually opposing objectives for uniform arrays since if the fixed number of elements is N then D and d are proportional to each other and related by

$$D = d(N-1) \quad (4.1.12)$$

In order to get the best of both requirements some non-uniform spacing scheme must be used. In addition the gain will be nonuniformly adjusted since it has been shown [42] that shading the gains, although it has no effect on the amplitudes of the grating lobes, can greatly reduce the size of the aperture side lobes.

The introduction of focusing into the uniform phased array does not alter the angle θ_g in equation (4.1.9). However, each major

lobe (including grating lobes) pinches down to the spot size s at the focal plane distance of F .

4.2 Phased Array Diffraction Suboptimization

The problem of maximizing the rolloff of an array diffraction pattern at the coverage region boundary subject to constraints on the main lobe, sidelobes and number of phased array elements is a familiar one in antenna theory. It has been recognized recently [54] that this problem is analogous to the problem of designing a so-called Finite Impulse Response (FIR) low pass digital filter [2,55,56,57]. Thus, progress in the design of low pass digital filters with minimum transition bandwidths subject to band-pass and stopband constraints has inspired interest in the possibility of optimizing the design of phased arrays using these techniques [58]. The most notable recent studies have employed the use of the method of Remez [60] to accomplish optimization in the mini-max or Chebyshev sense.

The output of a linear array antenna of uniform element spacing, d , and terminal gains a_m subject to an incoming plane wave of frequency f_0 at incidence angle θ is

$$s(t) = A G(\eta) \exp[j(2\pi f_0 t + \phi)] \quad (4.2.1)$$

where $G(\eta)$ is the far field antenna pattern

$$G(\eta) = \sum_{m=1}^M a_m \exp(jkmd\eta) \quad (4.2.2)$$

where $k = 2\pi/\lambda$ and $\eta = \sin \theta$.

By analogy, the output of an FIR digital filter with impulse response samples h_m when the input is a sinusoid at radian frequency ω is

$$y(t) = A H(\omega) \exp [j(\omega t + \phi)] \quad (4.2.3)$$

where $H(\omega)$ is the frequency response given by

$$H(\omega) = \sum_{m=1}^M h_m \exp (jm\omega T) \quad (4.2.4)$$

where $\omega = 2\pi f$ and T is the uniform temporal sampling period of the digital filter. It can be seen that the two problems are analogous with the correspondence

$$f = \eta d / (\lambda T) \quad (4.2.5)$$

and

$$h_m = a_m \quad (4.2.6)$$

The shading problem consists of selecting the a_m values to minimize the sidelobes.

It will be seen that in the near field this analogy begins to break down. Emphasis here is to be placed on the reference to the similarity between uniformly spaced arrays in the far field and uniformly sampled FIR digital filters. It will be shown that the correct extension of the analogy in the near field arrays is the study of nonuniformly sampled FIR digital filters. However, it will be of interest for the moment to consider just how robust the far

solution will be when applied to near field, focused and steered situations. It will turn out that certain of the side lobes (termed aperture side lobes) will be suppressed in a robust manner. However, another class of side lobes (termed grating side lobes) will present a problem when the array is steered whether focused or not.

The Remez exchange algorithm is a computationally efficient method for iteratively adjusting the gain values a_i in equation (4.2.2) to optimize the diffraction side lobe structure. The optimization is executed in the mini-max sense utilizing the properties of Chebyshev polynomials. The Remez method has been developed primarily for one dimensional arrays. Nevertheless, the method can be extended with some difficulty to two dimensional arrays directly, or by using the McClellan transformation [62].

The one dimensional Remez exchange algorithm solves the Chebyshev approximation problem by searching for the extremal "frequencies" (in the sense of the filter analog) of the best approximation to an ideal frequency characteristic. At the beginning of each iteration one has a set of $r + 1$ attempted "extremal frequencies" $\{f_k\}$. One then computes a generalized "best" Chebyshev polynomial fit with error values of known magnitude δ and alternating sign on the set $\{f_k\}$. Upon interpolating this polynomial its actual extrema are located at $\{f'_k\}$ and if they exceed δ a new iterate is attempted using the new extremal set $\{f'_k\}$ and the desired maximum error δ . Occasionally the iteration fails to converge in certain ill conditioned cases due to numerical errors.

It should be emphasized that the normal application of the Remez algorithm is for uniformly spaced elements. Actually, in this case the quality of the Remez array solution is closely approached by using the Kaiser gain weighting coefficients given in closed form by [55]

$$a_m = \frac{I_0\{\omega_\alpha[(\frac{M-1}{2}) - (M - (\frac{M-1}{2}))^2]^{1/2}\}}{I_0[\omega_\alpha(\frac{M-1}{2})]} \quad (4.2.7)$$

where $I_0(.)$ is the modified zeroth order Bessel function, m is the element subscript, M the total number of elements and ω_α is a parameter that permits a tradeoff between bandwidth and the tolerated side lobe structure. Obviously the closed form solution of the Kaiser gains is preferable to the Remez procedure, but this conclusion is valid only for uniformly spaced arrays. The Remez procedure (in suitably modified form) offers a method for handling the case of arrays with nonuniformly spaced delays.

One observation is pertinent before proceeding to a discussion of some of the computed design results. If in equations (4.2.1) and (4.2.2) each antenna element is phased by an additional amount given by $\exp(-jkmd\alpha)$ then the resulting "steered" diffraction pattern becomes

$$G_\alpha(\eta) = \sum_{m=1}^M a_m \exp(jkmd(\eta - \alpha)) \quad (4.2.8)$$

This action will simply shift the diffraction pattern to an effective steering angle ψ given by

$$\psi = \sin^{-1} \alpha \quad (4.2.9)$$

without altering the maximum values of the side lobe structure. Hence, array gains optimal in the far field are independent of the steering angle, ψ . This includes all side lobes, and specifically includes grating lobes. One of the important effects ignored in (4.2.8) is the fact that this will not be true with finite sized elements. The individual element radiation patterns will taper the interference pattern of equation (4.2.8) as will be seen in the next section.

Design of two dimensional FIR digital filters is an active research area. A good overview of the status of this research is given in the recent paper by Harris and Mersereau [70]. Just as in the one dimensional case an analogy may be drawn between the design of two dimensional FIR filters, frequency characteristics and two dimensional phased array farfield diffraction patterns. The two dimensional far field array diffraction pattern is given by

$$G(\eta, \mu) = \sum_{m=1}^M \sum_{n=1}^N a_{mn} \exp(jk[m\eta + n\mu]d) \quad (4.2.10)$$

where η and μ are direction cosines in the x and y direction respectively, $k = 2\pi/\lambda$ and d is the x-axis or y-axis element spacing.

The two dimensional frequency response of a two dimensional FIR digital filter is given by:

$$H(\omega_1, \omega_2) = \sum_{m=0}^M \sum_{n=1}^N h_{mn} \exp(j(m\omega_1 + n\omega_2)T) \quad (4.2.11)$$

where ω_1 and ω_2 are the two frequencies and h_{mn} the two dimensional impulse response of the filter. Such filters are extremely useful in image processing studies [70].

Again the analogy will be complete if one forms the association of

$$f_1 = \frac{\omega_1}{2\pi} = \frac{nd}{\lambda T} ; f_2 = \frac{\omega_2}{2\pi} = \frac{\mu d}{\lambda T} \quad (4.2.12)$$

and

$$h_{m,n} = a_{m,n} \quad (4.2.13)$$

One simple method of designing optimal two dimensional gains is to employ the McClellan transformation [62] which converts an optimal one dimensional FIR filter into a two dimensional rotationally symmetric filter. This has been done for a 16x16 array in Figure 4.2.2. This characteristic was obtained from a one dimensional filter transfer function corresponding to an axial section of Figure 4.2.1. Two dimensional array interference patterns with gain shading will also be shown in the next section.

4.3 Diffraction Pattern Simulations of Phased Arrays

We consider the Fresnel near field interference problem treated by Goodman [52] using the coordinate system shown in Figure 4.3.1. The phased array elements are considered to impress a signal on the medium in the source plane (x_1, y_1) while the resulting interference pattern is measured in the observation plane (x_0, y_0) .

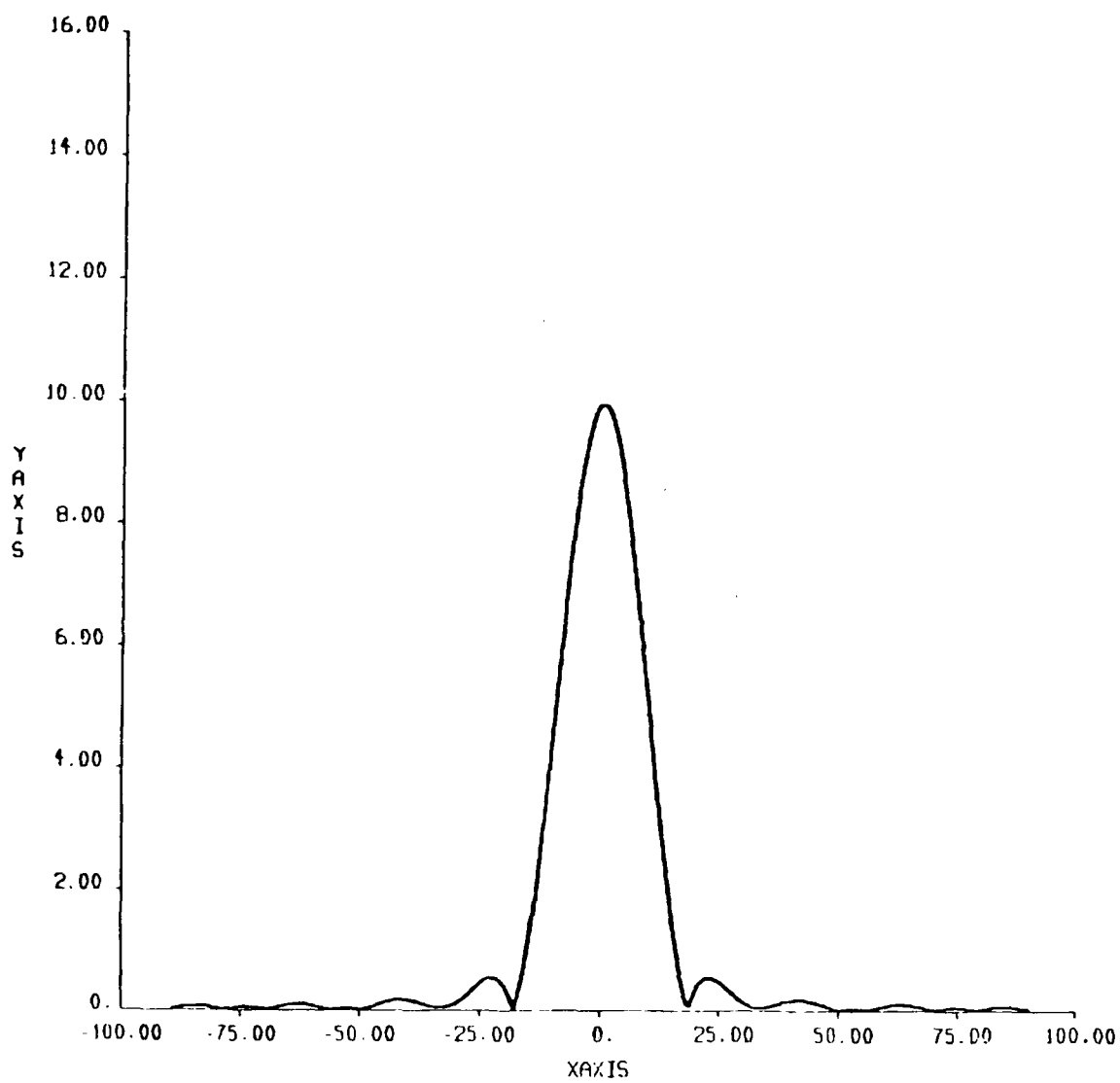


Figure 4.2.1 Beam Pattern; 16 Element Linear Array, Kaiser Gains, Far Field.

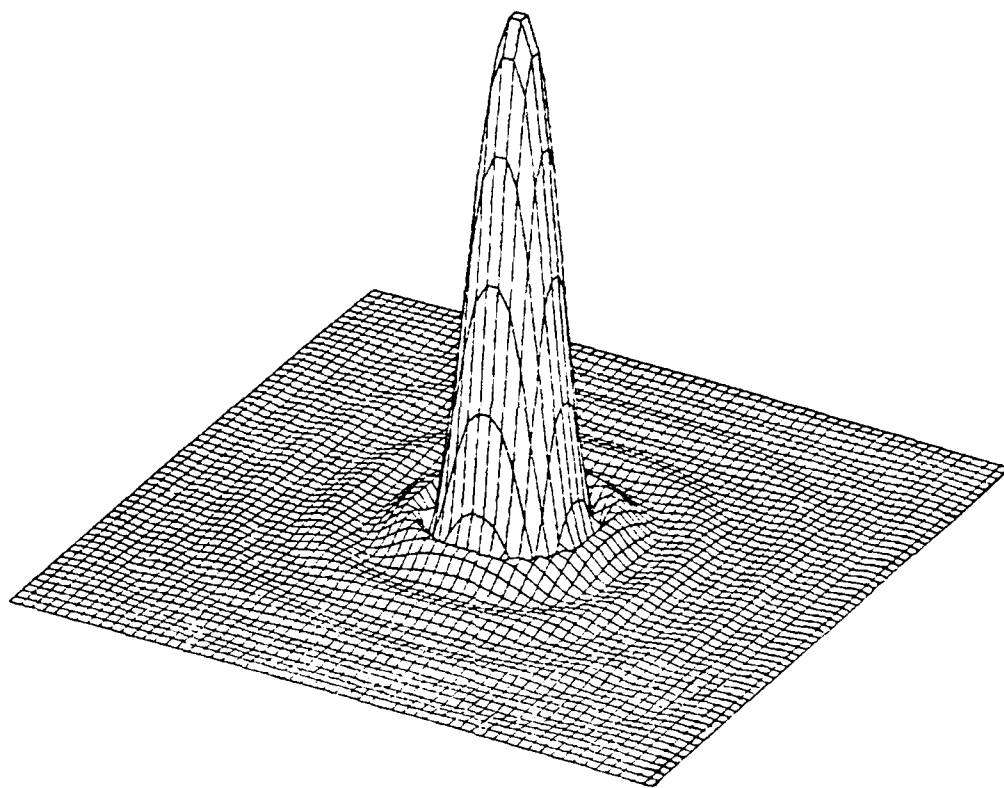


Figure 4.2.2 Beam Pattern; 16x16 Array, Kaiser Gains,
Far Field Mapped by McClellan Transformation.

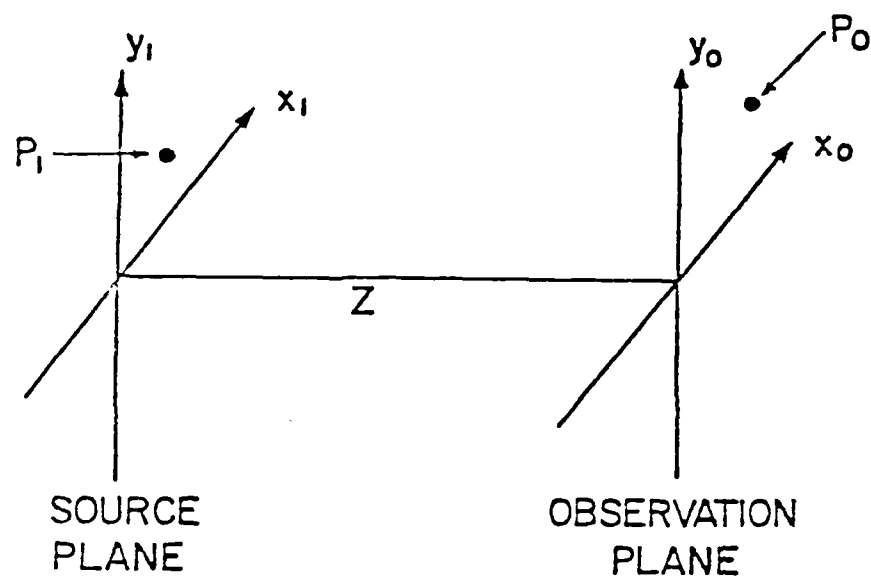


Figure 4.3.1 Coordinate System for Fresnel Diffraction Pattern.

Consider the source to be an array of transducers imposing identically shaped radiation patterns $G(x_1, y_1)$ shifted to locations x_{cm}, y_{cm} in the source plane. Denoting $U_1(x_1, y_1)$ as the ultrasonic field, we have

$$U_1(x_1, y_1) = \sum_{m=1}^M a_m e^{j\theta_m} G(x - x_{cm}, y - y_{cm}) \quad (4.3.1)$$

To obtain concrete results a Gaussian pattern is assumed for $G(x, y)$:

$$G(x, y) = e^{-1/2(x^2 + y^2)/L^2} \quad (4.3.2)$$

The results tend to be mildly independent of this assumed form for G except through the width parameter L . It is readily shown [61] that the resulting Fresnel region interference pattern is

$$U_0(x_0, y_0) = \frac{-jk}{z} e^{jkz} \left(\frac{1}{L^2} - \frac{jk}{z}\right)^{-1} \sum_{m=1}^M a_m e^{j\theta_m} e^{jg_m} \quad (4.3.3)$$

where

$$g_m = \frac{k}{2z} \left[1 - \frac{jkL^2}{z}\right]^{-1} [(x_0 - x_{cm})^2 + (y_0 - y_{cm})^2] \quad (4.3.4)$$

and where z is the source to observation plane perpendicular separation. This result has been computed for a variety of cases for both one and two dimensional arrays with uniform and nonuniform spacing. All results have been scaled up by a factor of z to make the figures readable. Also the diffraction calculations here assume no interelement coupling.

Techniques for suppression of sidelobes in the diffraction pattern described by equation (4.3.3) are primarily available in the

far field. In this section, we examine the robustness of these far field gains in the near field for a focused array. One can rewrite equation (4.3.3) explicitly in the far field limit for a cartesian array with M columns located at x_{cm} and N rows at y_{cn} as follows:

$$U_0(x_0, y_0) = (2\pi L^2)^{1/2} \exp(jkz + jk(x_0^2 + y_0^2)/(2z)) \sum_{m=1}^M \sum_{n=1}^N a_{mn} \exp(-j[2\pi/\lambda z](x_0 x_{cm} + y_0 y_{cn}) - \theta_{mn}) \quad (4.3.5)$$

Evans [54] notes the similarity between the gain optimization of Equation (4.3.4) and the design of optimum FIR filters. Using modified Kaiser gains [55] typical diffraction patterns have been computed for uniformly spaced arrays in the near field.

We begin by considering a one dimensional array with 16 elements with $D = 1\text{mm}$, and element separation of 2mm between centers. At 4.5 MHz the grating lobes are spaced at about 8.5° . Figure 4.3.2 gives the Fresnel near field interference pattern out to 0.6 meters for an unfocused array with uniform gains. Note that the main lobe has only just begun to appear among the many side lobes. Figure 4.3.3 is the same array steered by an angle of 3° . The grating lobe at -5.5° is quite apparent.

In Figure 4.3.4 the gains, a_i , are adjusted to the Remez optimum values for far field sidelobe suppression. Note that the near field shows substantial improvement in spite of the fact that the a_i are strictly optimal only in the far field and no focusing has been applied. Figure 4.3.5 shows the effect of the same

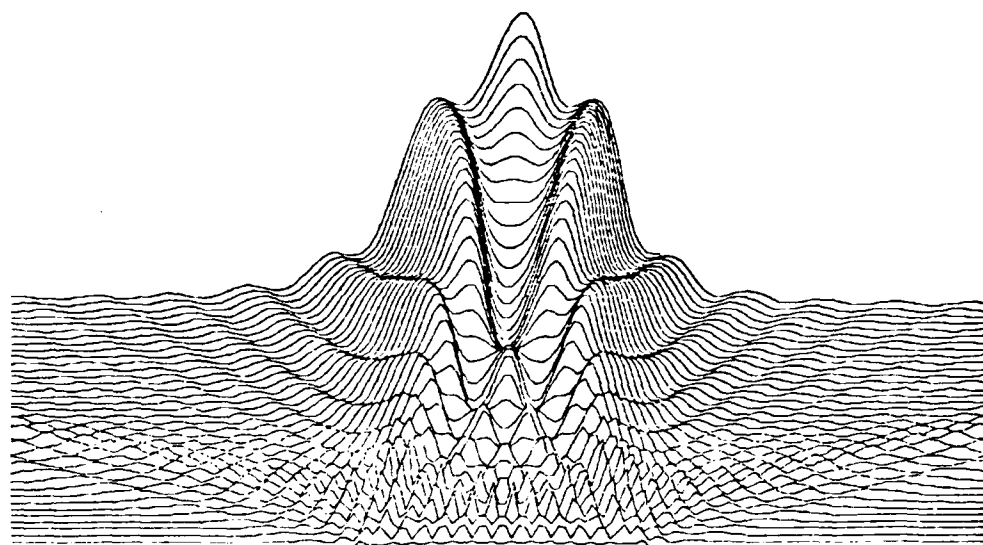


Figure 4.3.2 Beam Pattern; Uniform Gains, Unfocussed

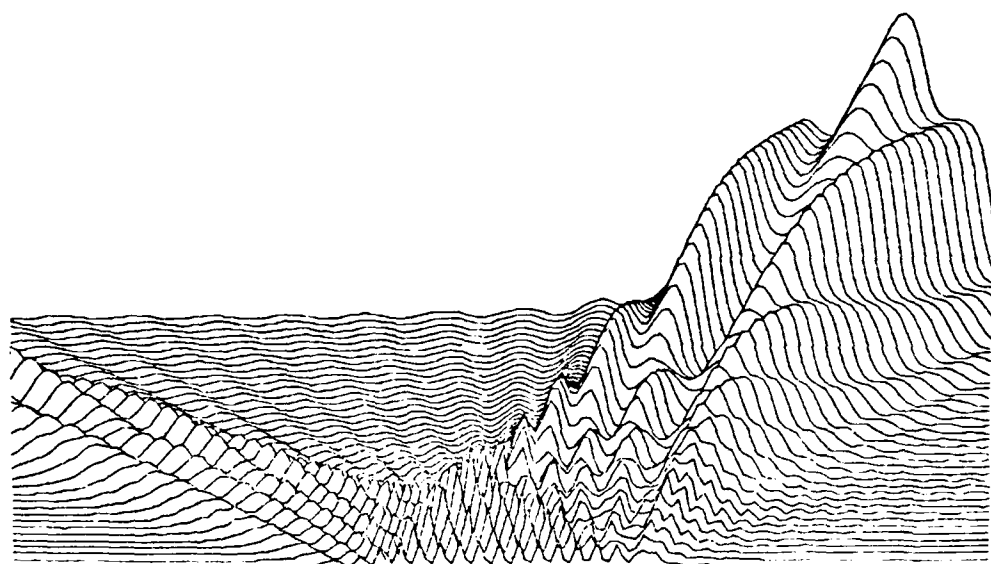


Figure 4.3.3 Beam Pattern; Uniform Gains, Unfocussed, Steered

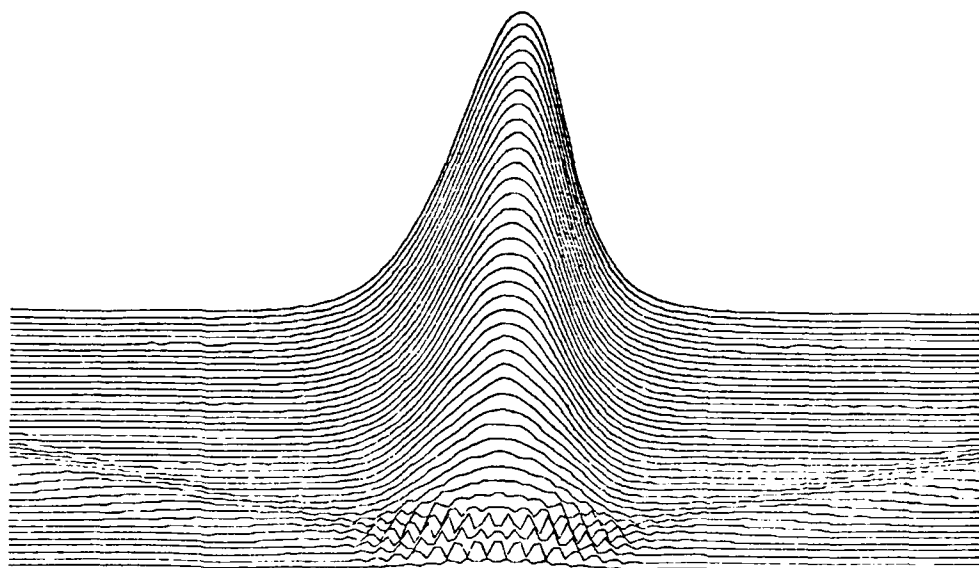


Figure 4.3.4 Beam Pattern; Kaiser Gains, Unfocused

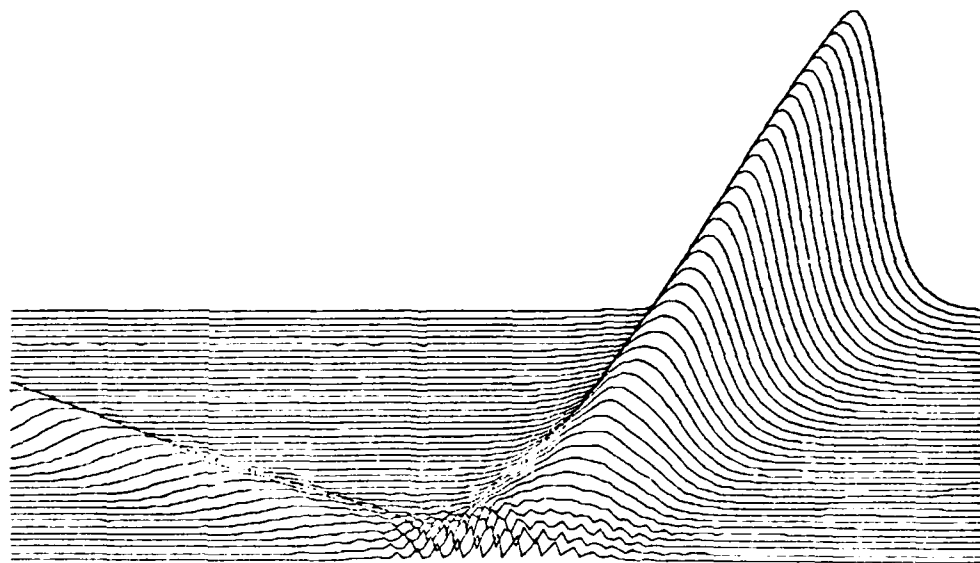


Figure 4.3.5 Beam Pattern; Kaiser Gains, Unfocused, Steered

optimizing gains with 3° of steering on the unfocused array. Obviously, aperture side lobes improve greatly as in the unsteered case, but grating lobes are not suppressed at all.

Figure 4.3.6 shows the resulting pattern when the array in Figure 4.3.5 is focused in the near field at .25 m. The focal spot is close to the optimum size dictated by equation (4.1.4) of 2.4 mm. The aperture side lobes are nearly unobservable on an unlogged plot. Finally in Figure 4.3.7 the focused and steered array with optimum far field gains shows a sharp focal spot but it also has the prominent grating lobe at -5.5° which can become larger than the main lobe as the grating lobe passes in front of the array with increasing steering angle. Note that the grating lobe shows a focusing effect also. In practice, the number of elements required to eliminate grating lobes due to dense spatial sampling may be unacceptably large. From these figures, it should be clear that far field optimum gains are quite effective in the near field unsteered case, whether focused or unfocused, in combating aperture sidelobes, but this approach by itself is ineffective in suppressing grating lobes.

In general, to be effective in achieving good lateral spatial resolution a phased array should be shaded in two dimensions since side lobes are significant in both directions. Computationally, the problem is the same as before, except the transducer locations (x_{cm}, y_{cn}) are distributed over a two dimensional mosaic in the source plane. Rewriting equation (4.3.3) with double subscripts for the Cartesian mosaic array, we have

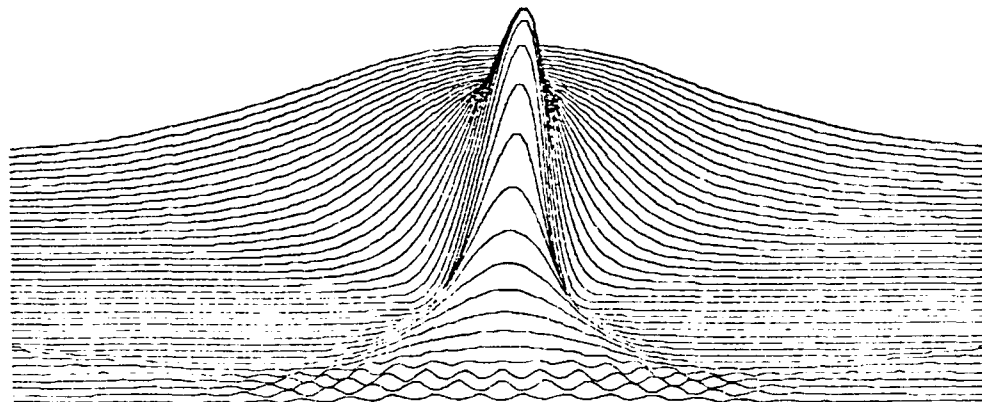


Figure 4.3.6 Beam Pattern; Kaiser Gains, Focused

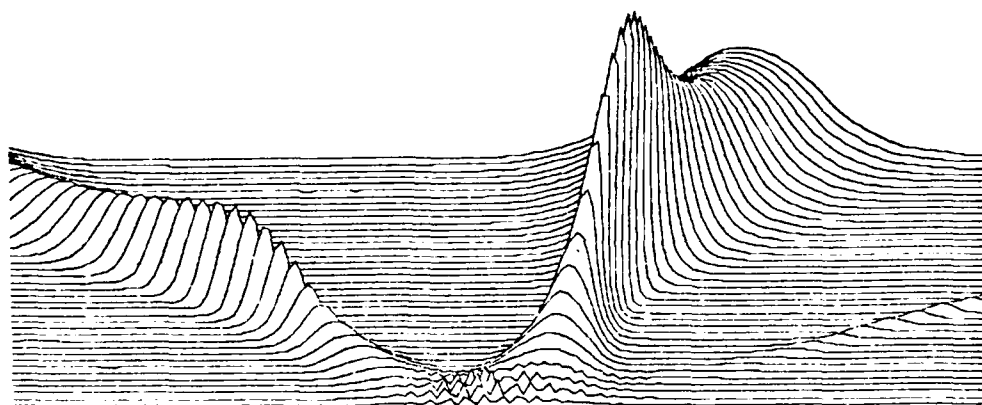


Figure 4.3.7 Beam Pattern; Kaiser Gains, Focused, Steered

$$U_0(x_0, y_0) = -\frac{jk}{z} e^{jkz} \left(\frac{1}{L^2} - \frac{jk}{z}\right)^{-1} \sum_{m=1}^M \sum_{n=1}^N a_{mn} e^{j\theta_{mn}} e^{jf_{mn}} \quad (4.3.5)$$

where

$$f_{mn} = \frac{k}{2z} \left[1 - \frac{jkL}{z}\right]^{-1} [(x_0 - x_{c_M})^2 + (y_0 - y_{c_N})^2] \quad (4.3.6)$$

The above formula is accurate provided Airy corrections are not needed. This condition will be guaranteed if [63],

$$z^3 \gg \frac{\pi}{4\lambda} [(x_0 - x_1)^2 + (y_0 - y_1)^2]_{\max}^2 \quad (4.3.7)$$

throughout the parameter ranges of interest. In other words we must have

$$\lambda \gg \frac{\pi}{4} z \tan^4 \gamma_{\max} \quad (4.3.8)$$

where γ_{\max} is the largest slant angle between points of interest in the source and receiver planes. However, this requirement is overly restrictive. For the Fresnel approximation to remain valid, it is only necessary for the Fresnel phase to dominate other higher order corrections. This is the so-called method of stationary phase.

To see what happens if the Fresnel field solution is used outside the allowed range for inequality in (4.3.7), the near field just in front of the radiators was computed. The result shown in Figure 4.3.8 clearly displays the Gaussian driver patterns for a two dimensional 5 x 5 array in the very near field.

Figure 4.3.9 illustrates a near field interference pattern for the aforementioned 5 x 5 element array. The gains are uniform

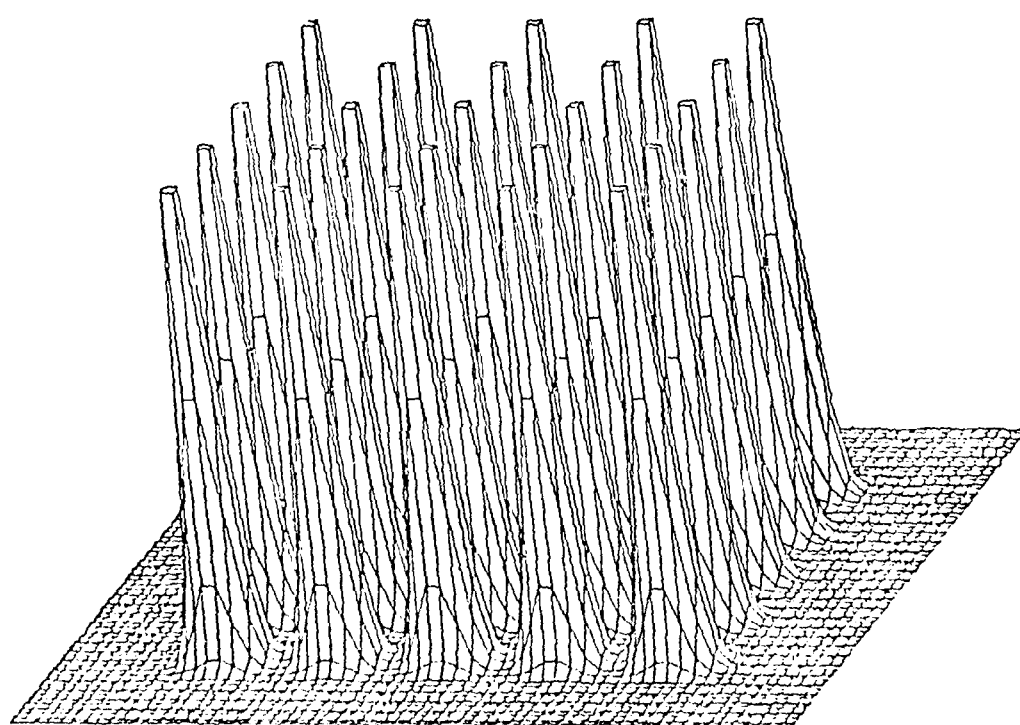


Figure 4.3.8 Fresnel Diffraction Pattern for 5×5 Array in the Very Near Field.

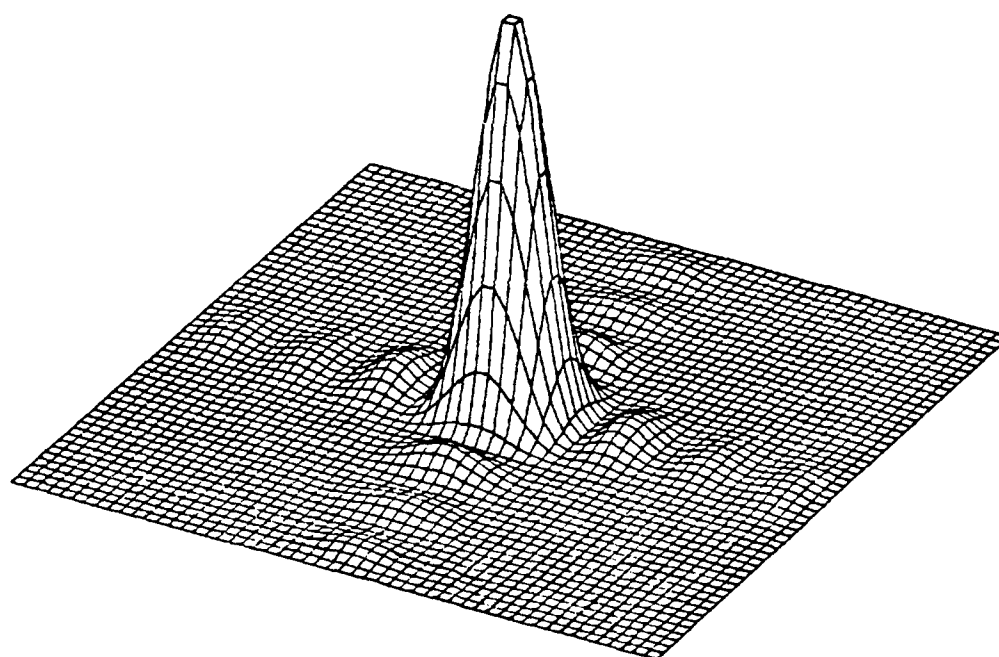


Figure 4.3.9 Beam Pattern; 5 x 5 Array, Uniform Gains,
Unfocused, $z = .8$ m.

($a_{mn} = 1$), the elements are of 1 mm circular diameter with 2 mm rectangular center to center spacing and the array is unsteered. Grating lobes are extremely small as a result of element directivity and lie just beyond the field of view. By steering the array 3° along both the x and y axes prominent grating lobes may be seen in Figure 4.3.10. The interference pattern is also asymmetric as a result of the local directivity of the radiating elements.

Side lobe structure is more easily visualized for focused arrays if the observation plane is moved closer to the array. Figure 4.3.11 illustrates the nearfield diffraction at $z = 0.125$ m for an unfocused, unsteered, unoptimized (uniform weight) 5×5 element array with 1 mm element diameter and 2.5 mm center spacing. By focusing the unsteered array in the center of the viewing plane at $z = 0.125$ m, we observe a substantially cleaner center spot even without gain optimization as shown in Figure 4.3.12. However, once again prominent grating lobes are observed. If the array is now steered without focusing or gain optimization the diffraction pattern is disrupted substantially in the very nearfield at $z = 0.125$ m. This is shown in Figure 4.3.13. Focusing at this steering angle improves the situation slightly as demonstrated by Figure 4.3.14.

A two-dimensional transformation from a one-dimensional Kaiser window is

$$a_{m,n} = \frac{I_0[\beta(1-(m^2 + n^2)^{1/2}/M)^{1/2}]}{I_0[\beta]} \quad (4.3.9)$$

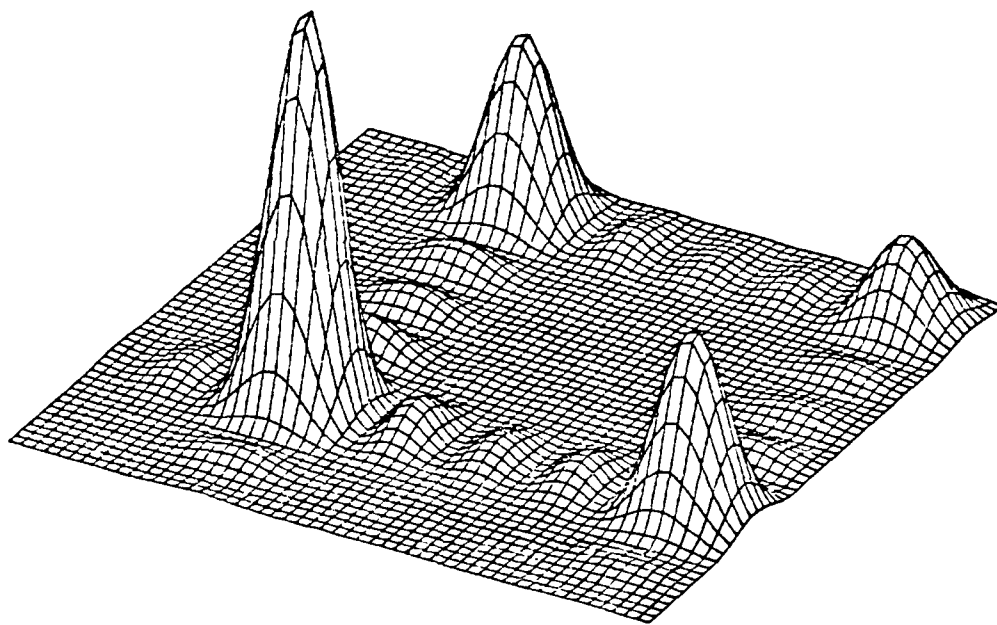


Figure 4.3.10 Beam Pattern; 5 x 5 Array, Uniform Gains,
Unfocused, Steered 3° , $z = .8$ m.

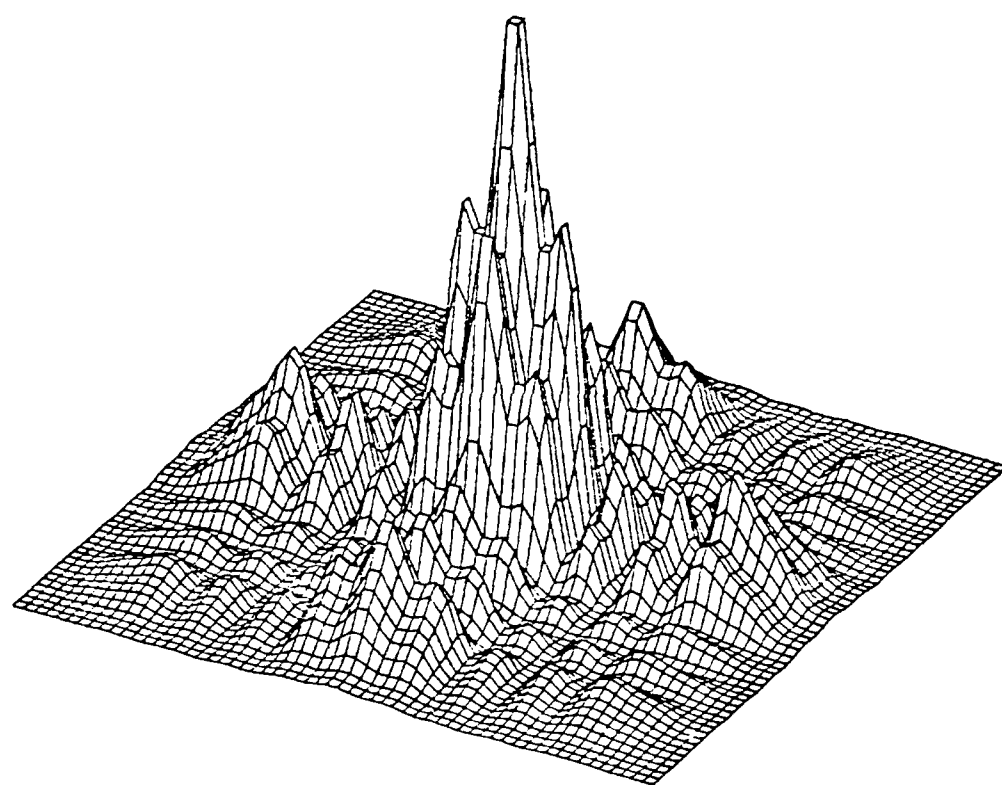


Figure 4.3.11 Beam Pattern; 5 x 5 Array, Uniform Gains,
Unfocused, $z = .125$ m.

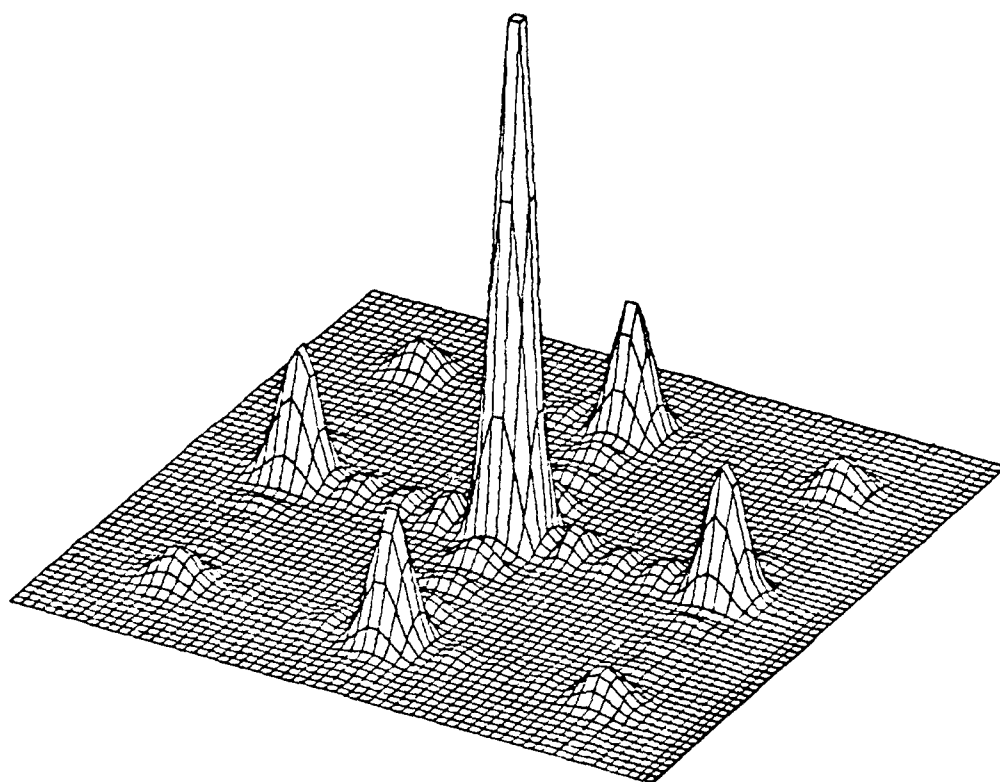


Figure 4.3.12 Beam Pattern; 5 x 5 Array, Uniform Gains,
Focused, $z \approx .125$ m.

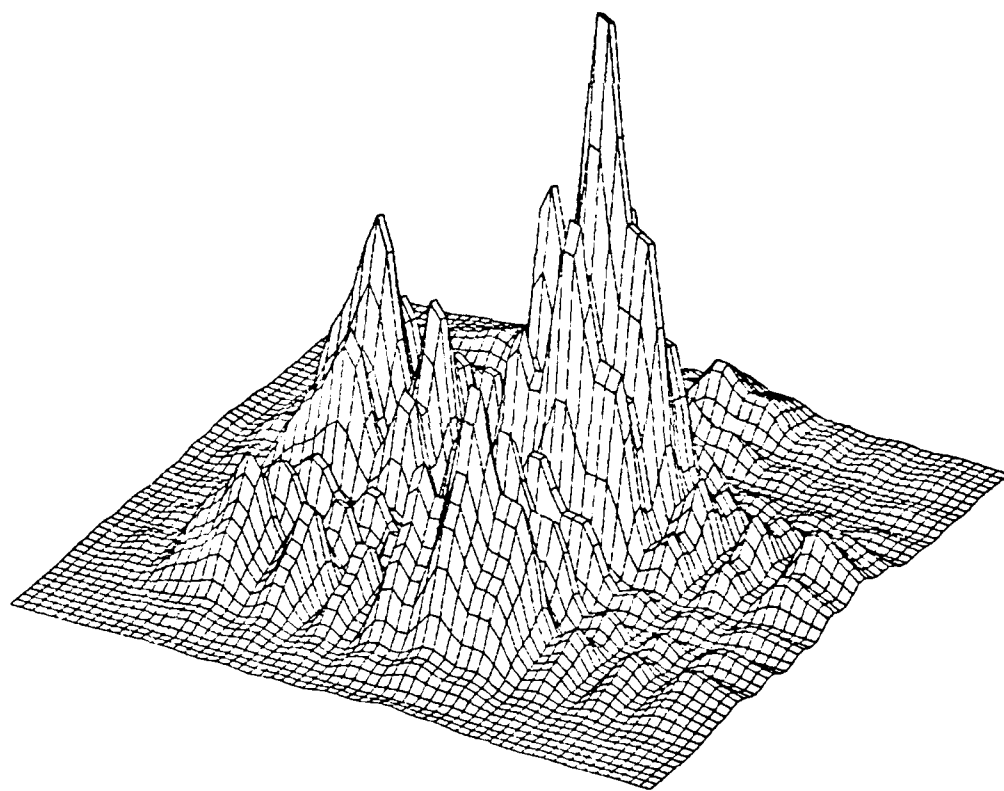


Figure 4.3.13 Beam Pattern; 5 x 5 Array, Uniform Gains,
Unfocused, Steered $\sim 5^\circ$, $z = .125$ m.

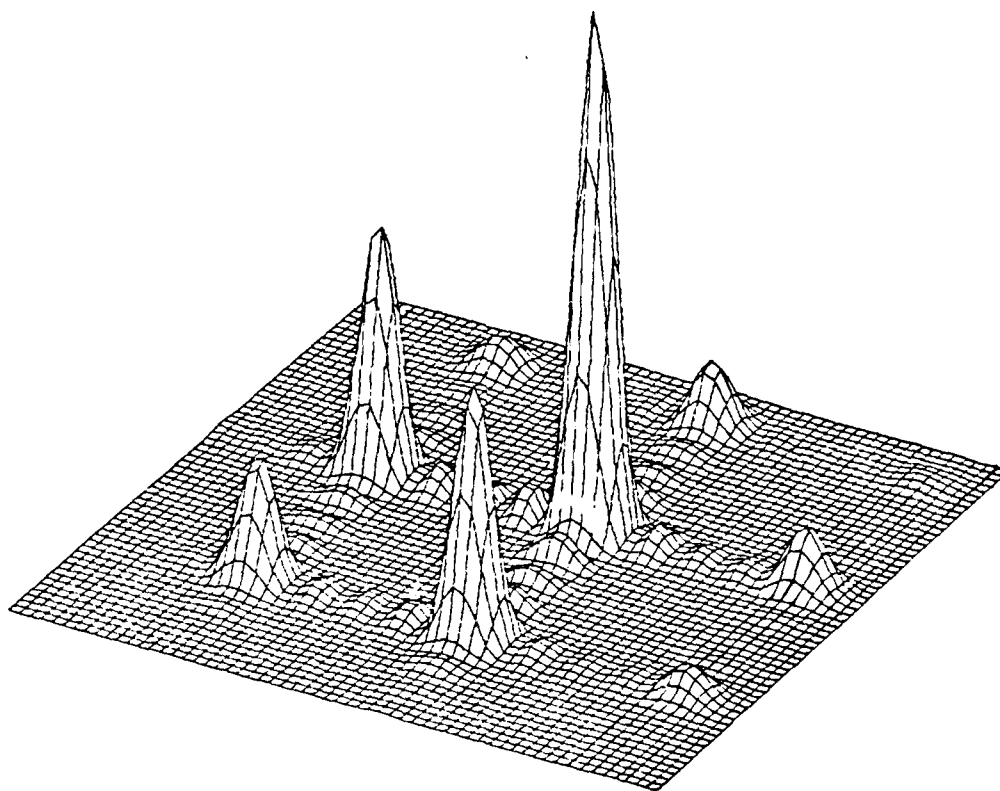


Figure 4.3.14 Beam Pattern; 5 x 5 Array, Uniform Gains,
Focused, Steered 2.5° , $z = .125$ m.

As noted previously, the Kaiser weights are nearly as effective as those produced by the Remez algorithm in the farfield. The effect is still dramatic even at $z = 0.125$ m in the nearfield as shown in Figure 4.3.15 where the Kaiser weights are used in the unsteered, unfocused counterpart to Figure 4.3.11. The small improvement due to focusing with Kaiser weights is shown in Figure 4.3.16. Nevertheless, the improvement in the Kaiser weighted equivalent to Figure 4.3.12 with steering and focusing (shown in Figure 4.3.17) is marginal, again due to grating lobes.

4.4* Grating Lobe Suppression by Nonuniform Spacing

In addition to these basic focusing considerations, there remains the problem of diffraction side and grating lobe suppression. As we have seen these are extremely disruptive in both the near field and far field even though optimal farfield weights greatly reduce the aperture side lobes in the nearfield. Further improvements can be expected with a Remez algorithm designed explicitly for nearfield optimization. The angular spacings of these lobes in uniform arrays were given in equations (4.1.9) and (4.9.10). For a nonuniform array the positions and amplitudes of aperture and grating lobes must be computed numerically. In many situations it may not be feasible or desirable to make d sufficiently small to eliminate the grating lobes from the field of view. It is clear from the cases presented above that the normal pure tone continuous wave (cw) grating lobes are not reduced by Remez or Kaiser shading. In terms of the

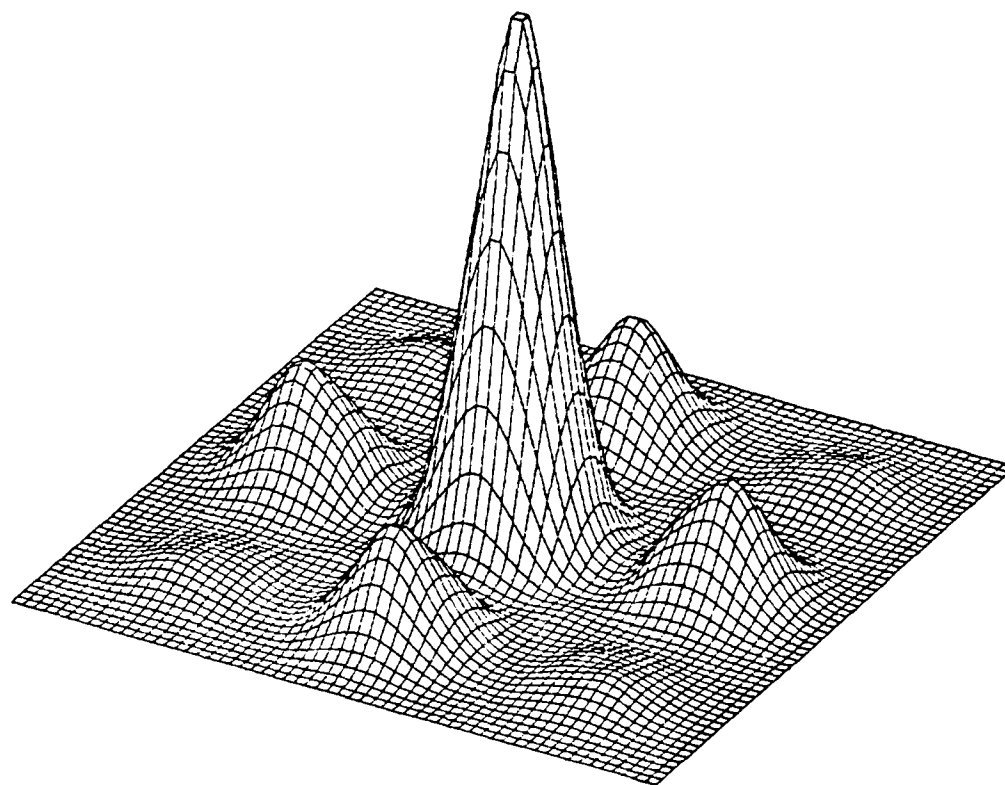


Figure 4.3.15 Beam Pattern; 5 x 5 Array, Kaiser Gains,
Unfocused, $z = .125$ m.

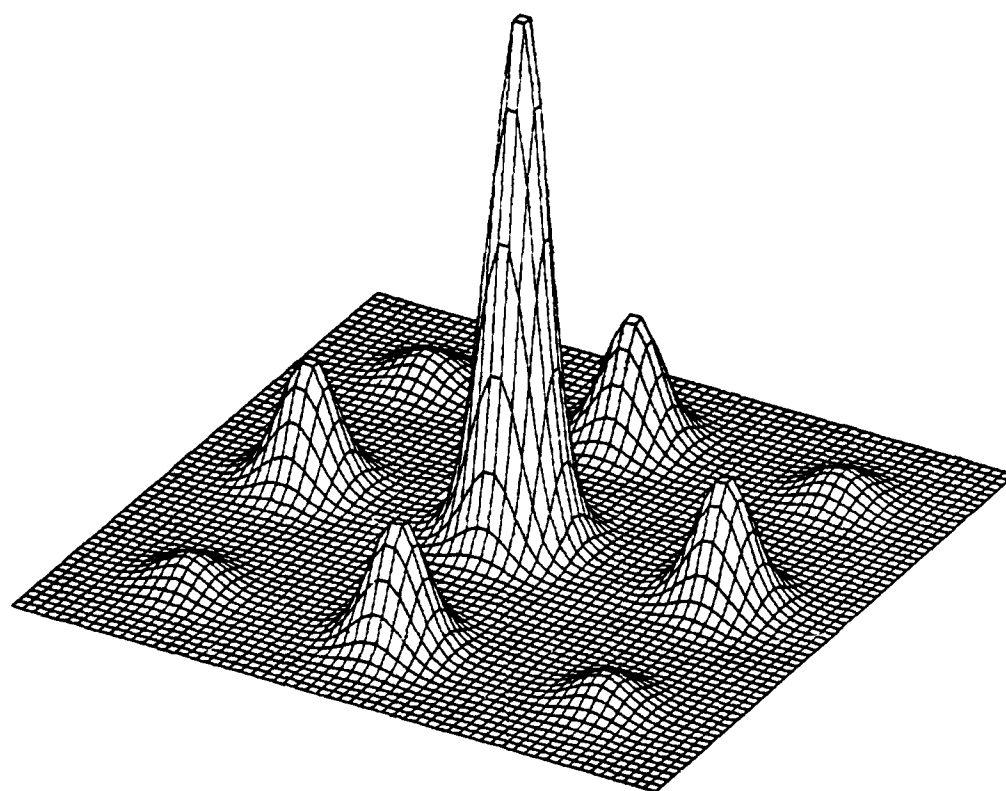


Figure 4.3.16 Beam Pattern; 5 x 5 Array, Kaiser Gains,
Focused, $z = .125$ m.

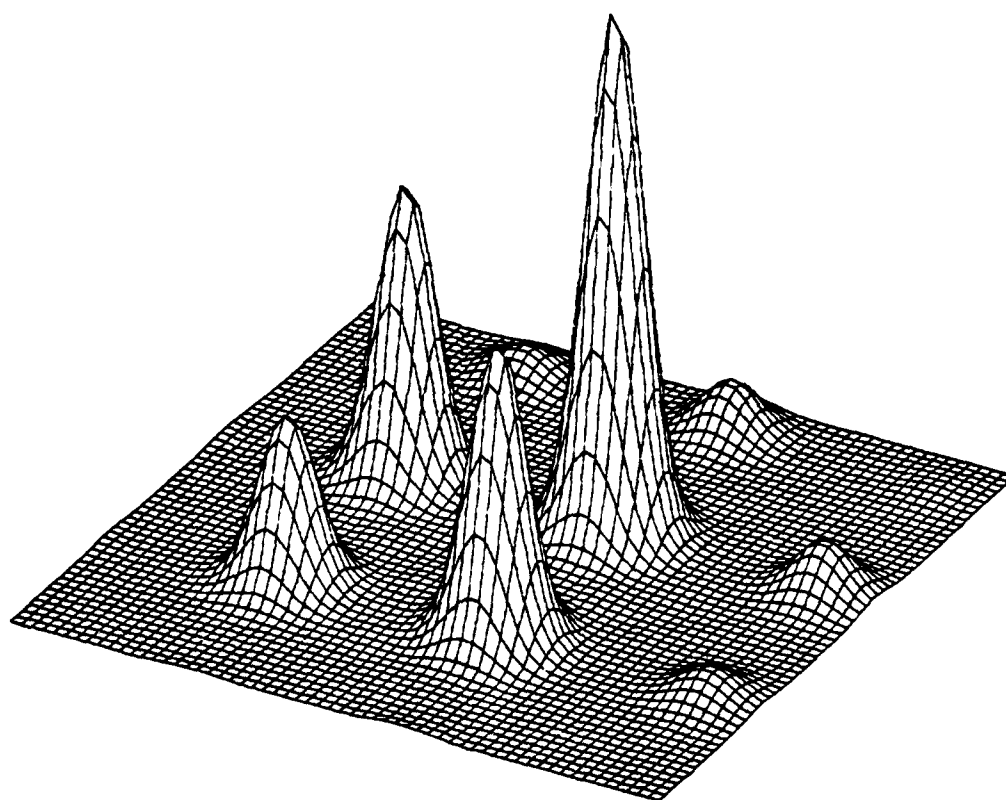


Figure 4.3.17 Beam Pattern; 5 x 5 Array, Kaiser Gains,
Focused, Steered 2.5° , $z = .125$ m.

uniformly sampled discrete FIR filter analogy the grating lobes are similar to frequency aliasing in an undersampled digitized signal [2]. The grating lobes are replicas of the main lobe and hence cannot be manipulated independently under this analogy.

There are at least three techniques to cope with the grating lobe difficulties:

1. Subarray Constrained Feed Systems and Aperiodic Arrays

In these systems the elements are not phased with single linear or quadratic phase tapering, but rather with an admixture of such phases [65,66,67] to give a random or aperiodic effect. Similar results can be obtained with nonuniformly spaced arrays.

2. Wideband Signal Design

In this approach the phased array is designed to jointly optimize spatial and temporal resolution by utilizing diffraction coherently across a wide band of frequencies [68,69]. This technique may not be applicable for quasioptical applications or distributed targets.

3. Optical Methods

In this method an ultrasonic lens system demagnifies an oversized array (obviously without changing λ) so as to decrease the effective element spacing. The resulting image then projects grating orders at a wider angle while improving the individual active element radiating pattern.

The solution used here and explained in the following section is a variation of the first technique. The array gains, spacings and phases are all adjusted simultaneously so as to most effectively reduce the size of the grating lobes. As an initial step in this direction it was decided to observe some nonuniform element spacing schemes alone for their affect on the beam patterns.

As noted in the previous section, the grating lobe problem cannot be reduced by element gain shading alone. In fact, this has no effect on the size of the grating lobes. It has been known for several decades that a solution for this problem is to use nonuniform element spacings. In an early study by King, et al. [40] several ad hoc spacings were compared. These included the arithmetic progression series, the elimination of multiple spacing series and the logarithmic series. The elimination-of-multiple-spacing series is the best of these ad hoc schemes.

Simulations of an 11×11 two dimensional, unshaded, focused array formed by a uniform and nonuniform Cartesian element spacing array are shown in Figure 4.4.1 and Figure 4.4.2. These show a remarkable improvement for grating lobe reduction for such a small array. Essentially the grating lobes have been almost totally pushed down and spread to outside of the viewing window of the figure. The Cartesian elimination-of-multiple spacing grid used to obtain these results is shown in Figure 4.4.3. The departure from uniform spacing required to obtain these results is surprisingly small. Figure 4.4.4 and Figure 4.4.5 show typical element spacings

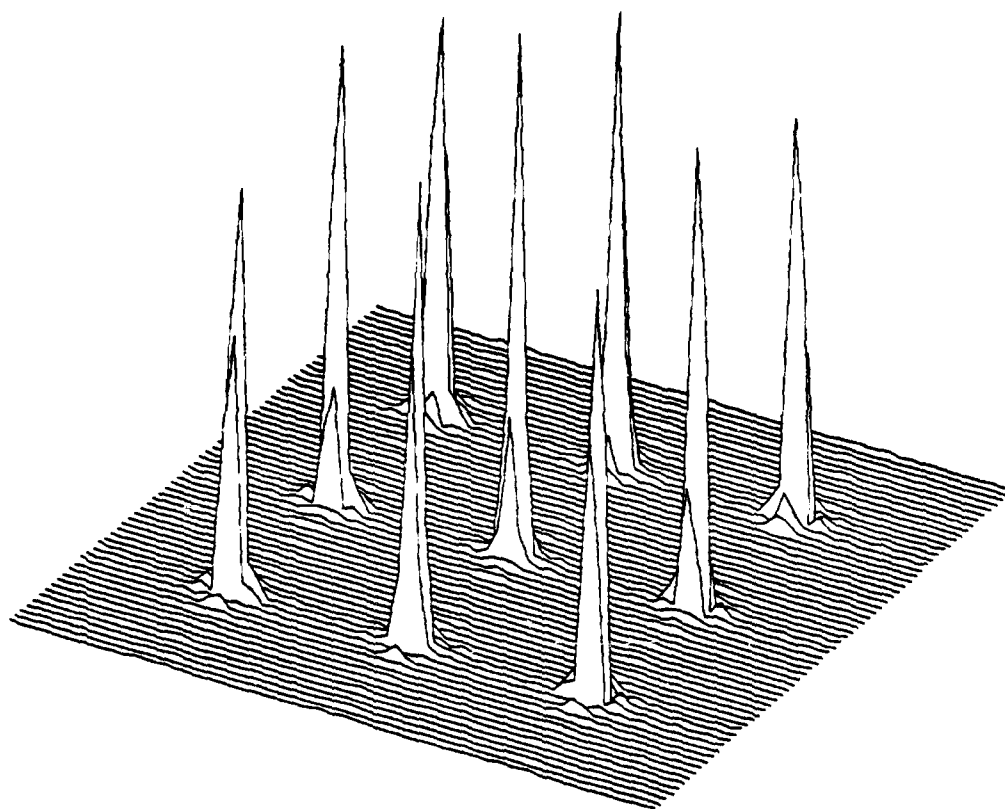


Figure 4.4.1 Beam Pattern; 11 x 11 Array, Uniform Gains, Focused, Uniform Spacings.

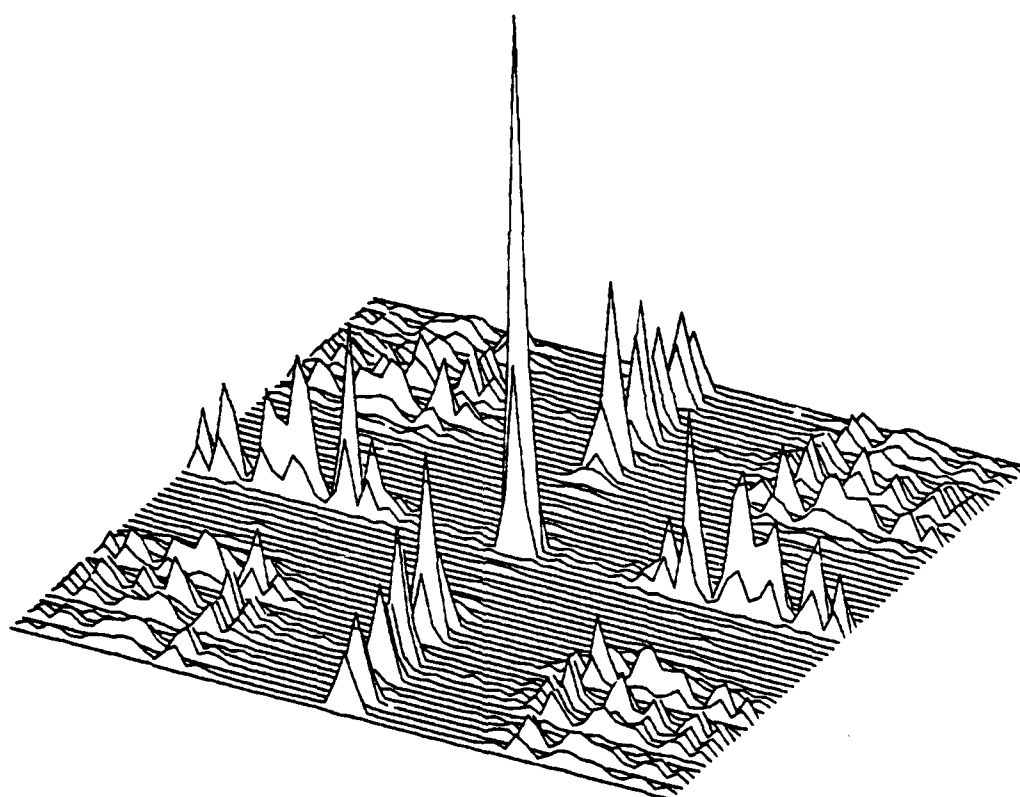


Figure 4.4.2 Beam Pattern; 11 x 11 Array, Uniform Gains, Focused, Elimination of Multiple Spacing.

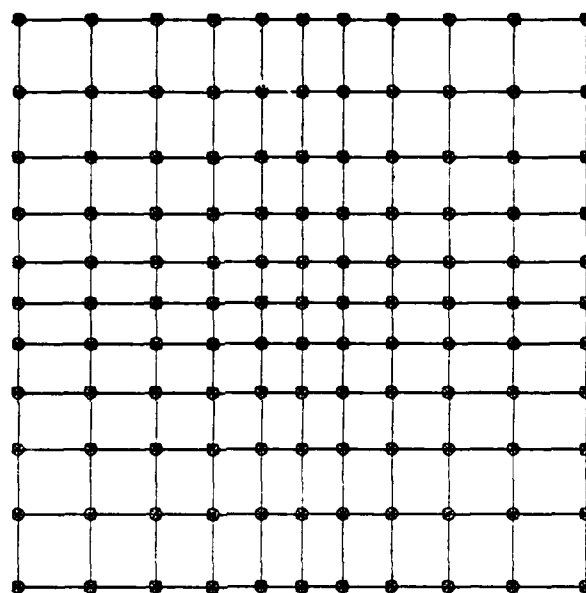


Figure 4.4.3 Planar Array; 11 x 11 Elements,
Elimination of Multiple Spacing.

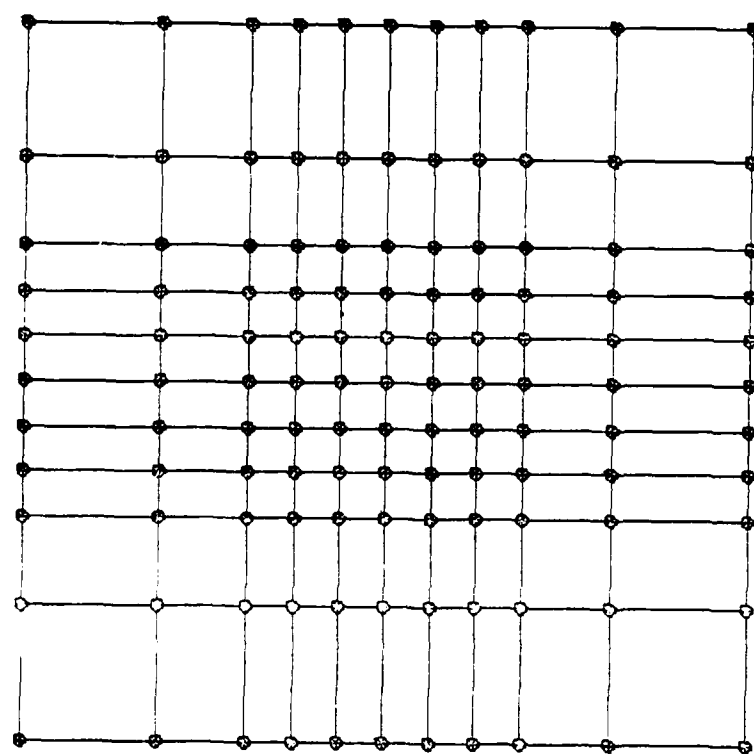


Figure 4.4.4 Planar Array; 11 x 11 Elements, Arithmetic Progression Spacings.

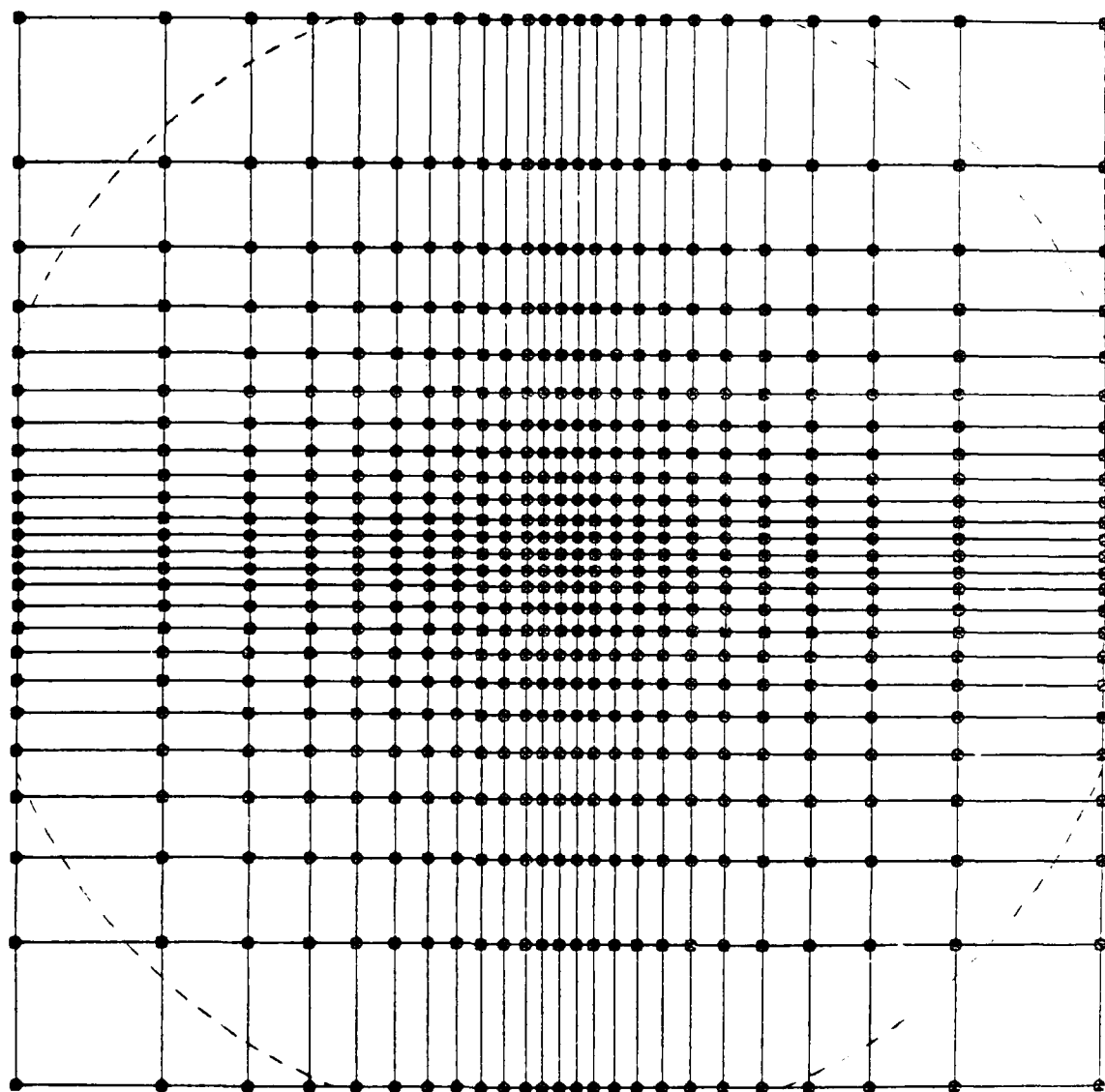


Figure 4.4.5 Planar Array, 25 x 25 Elements, Logarithmic Spacings.

of two dimensional arrays using arithmetic progression and logarithmic schemes. These are included for references and completeness even though the figures shown are not all for the same array scale. The dashed line on Figure 4.4.5 shows the elements which could be left out of the array with virtually no effect on the beam pattern but with a resultant 28 array element savings.

4.5 Minimax Beam Pattern Optimization of Nonuniformly Spaced Cartesian Arrays

In general there are three sets of parameters which can be adjusted to optimize the nonuniformly spaced array. These are the spacings, gains and phases of the individual elements (the focusing phase is determined by the distance of an element from the center of the array plus a small optimizing adjustment). Several programs exist for nonlinear minimax optimization with respect to the three classes of parameters, \underline{V} , in this problem. A particularly useful technique devised by Madsen [41] has been employed to carry out this optimization. This approach may be formulated as minimizing

$$F(\underline{V}) = \max_{mn} |U_0(x_m, y_n) - U_D(x_m, y_n)| \quad (4.5.1)$$

where $1 \leq m \leq N$, $1 \leq n \leq N$, $\underline{V} = [v_1, v_2, \dots, v_p]$, and $U_0(x_m, y_n)$ is the desired array response. In general $P \ll N^2$ so the solution tends to involve the minimax solution of overdetermined systems of nonlinear equations.

The procedure requires the specification of an explicit expression for the linear derivatives of the objective function in

equation (4.3.3) for use in a method similar to a Newton Raphson iteration except with step sizes adjusted with each iteration to insure that the actual error reductions are close to those predicted by a locally linearized objective function. Typical forms for the derivatives with respect to the element gains (a_{mn}), locations (x_{cm}, y_{cn}) and phases (θ_{mn}) are given by the following expressions for a Cartesian grid array.

$$\begin{aligned} \frac{\partial}{\partial \theta_{pq}} |U_0(x,y)| &= K \{ [- \sum_{mn} R_{mn} \cos \phi_{mn} R_{pq} \sin \phi_{pq}] \\ &+ [\sum_{mn} R_{mn} \sin \phi_{mn} R_{pq} \cos \phi_{pq}] \} \end{aligned} \quad (4.5.2)$$

$$\begin{aligned} \frac{\partial}{\partial x_{cp}} |U_0(x,y)| &= kK(x-x_{cp})/(z^2 + k^2 L^4) \\ &\{ [\sum_{mn} R_{mn} \cos \phi_{mn} R_{pn} (z \sin \phi_{pn} + KL^2 \cos \phi_{pn})] \\ &+ [\sum_{mn} R_{mn} \sin \phi_{mn} R_{pn} (-z \cos \phi_{pn}) + kL^2 \sin \phi_{pn}] \} \end{aligned} \quad (4.5.3)$$

$$\begin{aligned} \frac{\partial}{\partial a_{pq}} |U_0(x,y)| &= K \{ [\sum_{mn} a_{mn} r_{mn} \cos \phi_{mn}] r_{pq} \cos \phi_{pq} \\ &+ [\sum_{mn} a_{mn} r_{mn} \sin \phi_{mn}] r_{pq} \sin \phi_{pq} \} \end{aligned} \quad (4.5.4)$$

where

$$\begin{aligned} K &= kL^2 [k^2 L^4 + z^2]^{-1/2} \{ [\sum_{mn} R_{mn} \cos \phi_{mn}]^2 \\ &+ [\sum_{mn} R_{mn} \sin \phi_{mn}]^2 \}^{-1/2} \end{aligned} \quad (4.5.5)$$

$$R_{mn} = a_{mn} r_m s_n \quad (4.5.6)$$

$$r_m = \exp \left[\frac{-k^2 L^2}{2z^2 + 2k^2 L^4} (x - x_{cm})^2 \right] \quad (4.5.7)$$

$$s_n = \exp \left[\frac{-k^2 L^2}{2z^2 + 2k^2 L^4} (y - y_{cn})^2 \right] \quad (4.5.8)$$

$$\phi_{mn} = \theta_{mn} + \frac{kz}{2z^2 + 2k^2 L^4} [(x - x_{cm})^2 + (y - y_{cn})^2] \quad (4.5.9)$$

Slight changes occur in equation (4.5.3) for the spatial derivative when the array is not constrained to be Cartesian.

Optimization of uniformly spaced array gains can be performed by methods analogous to those used in FIR filter design. These techniques alone, however are ineffective in suppressing grating lobes. Now it will be demonstrated that joint optimization of array element spacings, gains and phases offers an effective means of reducing grating lobes and handling near field behavior explicitly by solving nonlinear equations.

Minimax optimization of Cartesian arrays has been successful for both one- and two-dimensional arrays. A particularly useful technique has been to start out a given design with one of the several ad hoc nonlinear spacings studied by King et al. [40], and either a set of uniform gains or modified Kaiser gains [55]. No attempt was made to optimize the phase at this point to observe the benefits of this procedure because of the extra computational burden of these additional parameters to be optimized.

Figure 4.5.1 shows the beam pattern of a uniformly spaced linear 16 element array. Note the large grating lobes on each side of the main lobe. Figure 4.5.2 is the beam pattern for the optimized linear 16 element array of Figure 4.5.1. The large grating lobes have been substantially reduced with the half power width of the main beam being widened only slightly.

Similarly, Figure 4.5.3 shows the beam pattern of a uniform 11 x 11 element two dimensional planar array. Figure 4.5.4 is the beam pattern of the minimax optimized array.

4.6 Minimax Beam Pattern Optimization of Fresnel Lens Arrays

The Fresnel zone plate lens is a special antenna array configuration where the elements are annular with proper radii and widths which are arranged concentrically [71]. The radii of the rings are given approximately by

$$\rho_n = \sqrt{n\lambda F} \quad (4.6.1)$$

where F is the principal focal depth of the lens. Actually the full zone plate (with all of its rings present) acts simultaneously as a positive and negative lens with multiple "virtual" focal points at axial positions given by the distances [72]

$$z_m = \frac{F}{m} \quad [m = 0, \pm 1, \pm 2, \dots] \quad (4.6.2)$$

away from the lens plane. If the positive values of z_m correspond to positions in front of the lens then the negative virtual foci produce ring-shaped side lobes at off-axial positions in any plane

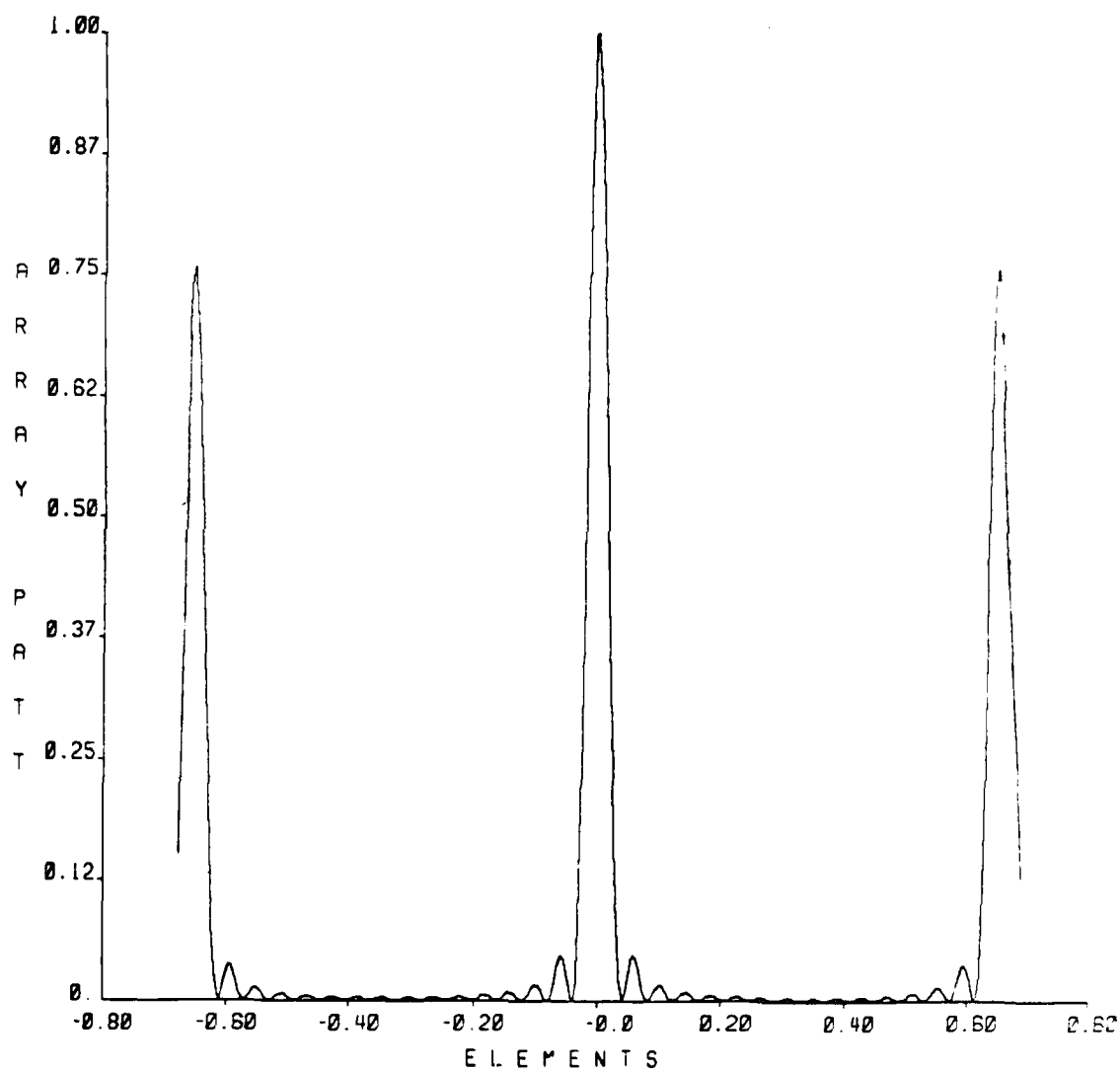


Figure 4.5.1 Beam Pattern; 16 Element Linear Array, Uniform Gains, Focused, Uniform Spacings.

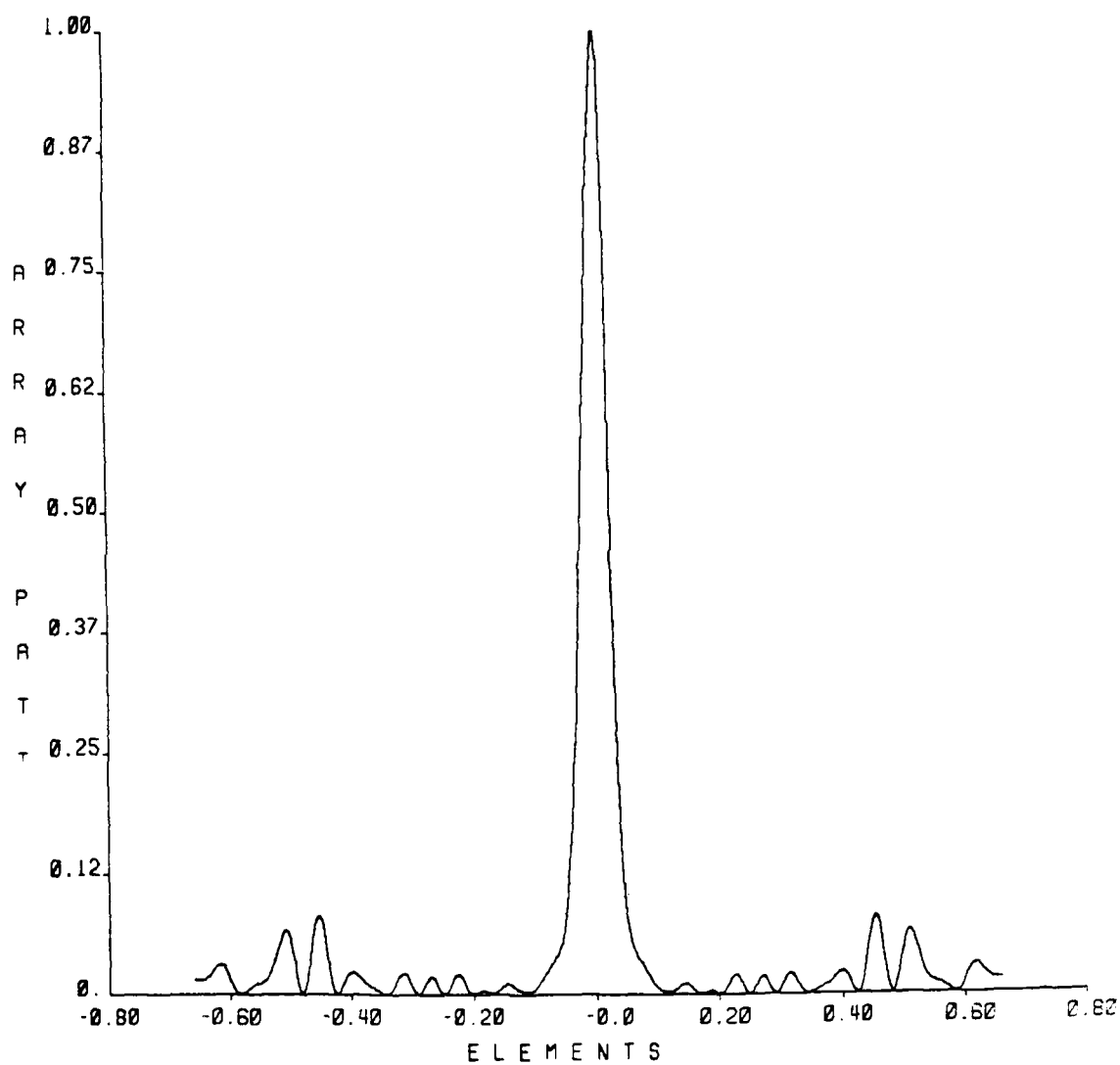


Figure 4.5.2 Beam Pattern; 16 Element Linear Array, Optimized Gains, Phases and Spacings.

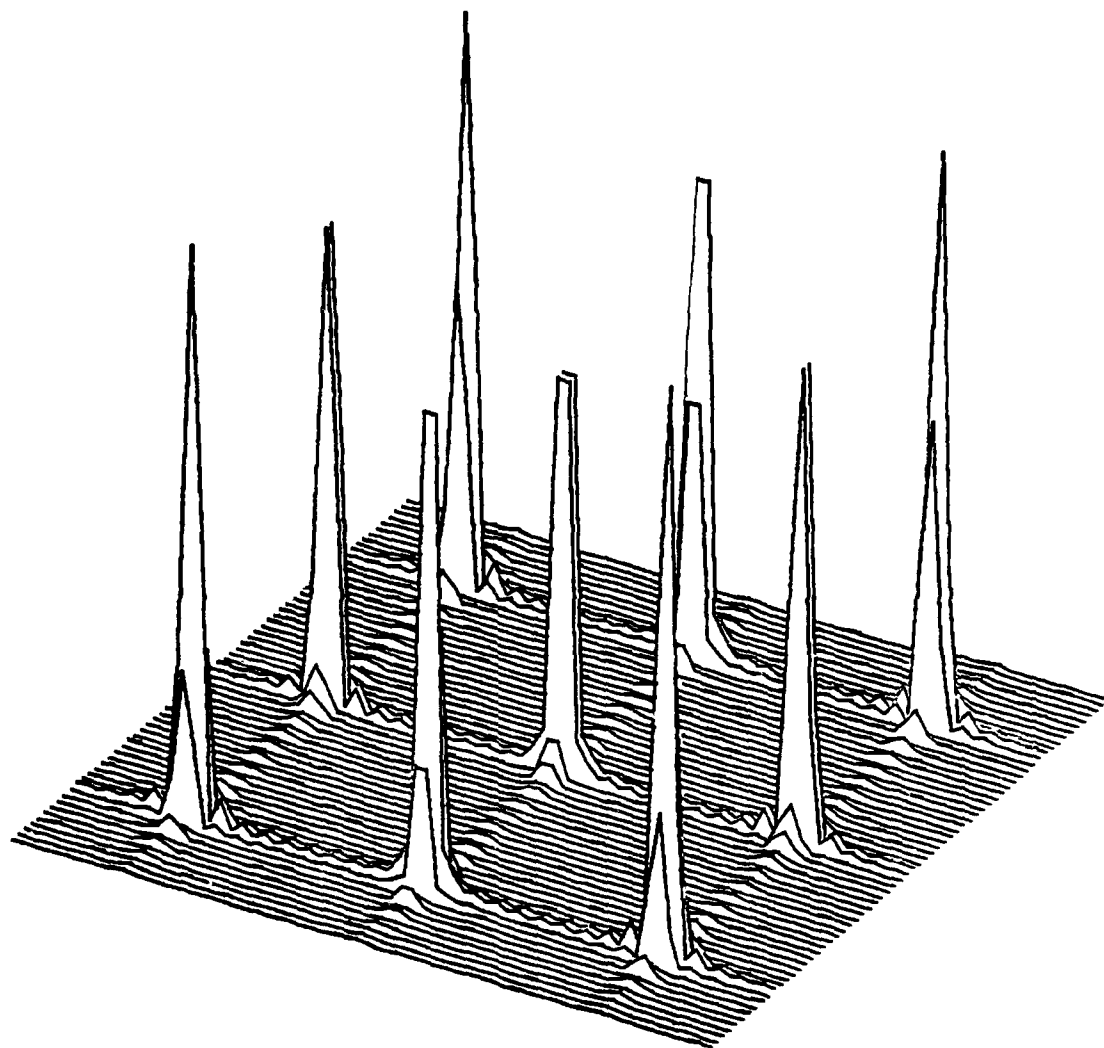


Figure 4.5.3 Beam Pattern; 11 x 11 Array, Uniform Gains,
Focused, Uniform Spacings.

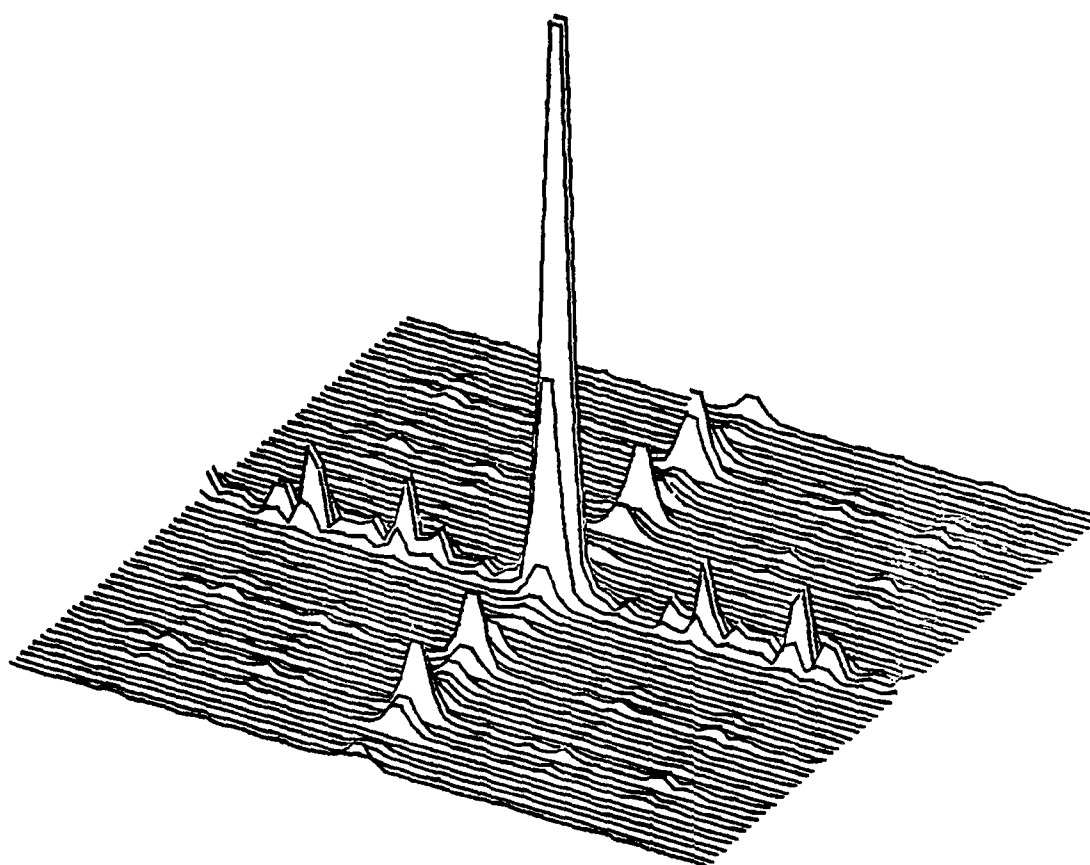


Figure 4.5.4 Beam Pattern; 11 x 11 Array, Optimized Gains,
Phases and Spacings.

perpendicular to the axis of the lens. These virtual spots and rings are analogous to grating lobes in unfocused arrays since the ring spacings of equation (4.6.2) undersample the aperture of the plate. This is illustrated in Figure 4.6.1 where the diffraction pattern is shown in the principal focal plane of a 10 element lens. As observed by other authors [73] the energy in these ring-shaped side lobes can be substantial.

Similarly, if one examines the virtual foci along the axis as shown in Figure 4.6.2 for a 14 element lens their effect can be equally unacceptable if some of these foci happen to lie inside the volume to be examined. If the zone plate focus is dynamically adjusted by varying λ or introducing additional "steering" phases for each ring then the virtual foci will also move.

To compensate for these undesirable phenomena one can resort to antenna array optimization. By varying the spacing of the rings a given region of the zone plate diffraction pattern can be swept free of strong virtual lobes. The method of nonuniform ring spacing is in contrast to the techniques discussed by Wild [77], Vilkomerson [78] and Burckhardt, Grandchamp and Hoffman [79] which can also be employed in conjunction with spacing adjustment where a single discrete focal spot is desired.

The approach chosen here was to adjust the spacings of the rings and add some shading to the element gains. No phases were included or optimized here. Using the unoptimized 14 element lens of Figure 4.6.2, the resulting optimized axial diffraction pattern

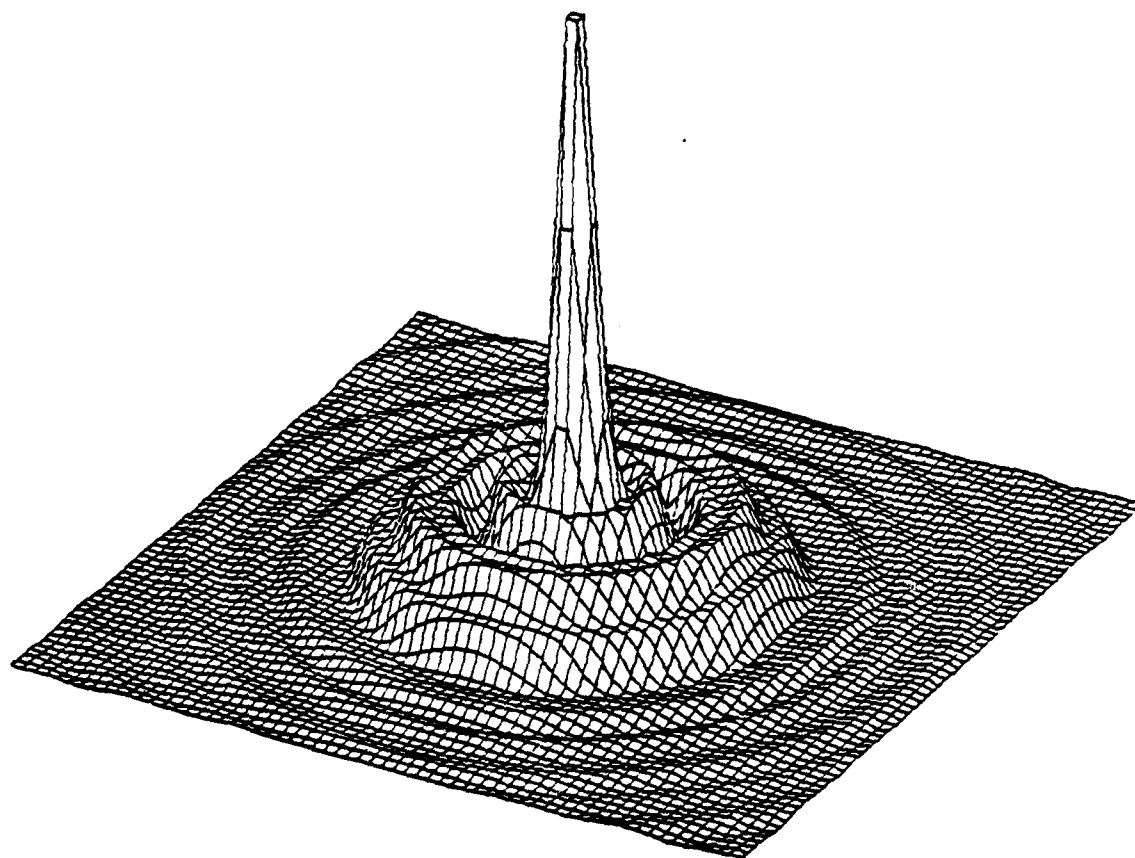


Figure 4.6.1 Fresnel Zone Plate Diffraction Pattern in the Principal Focal Plane.

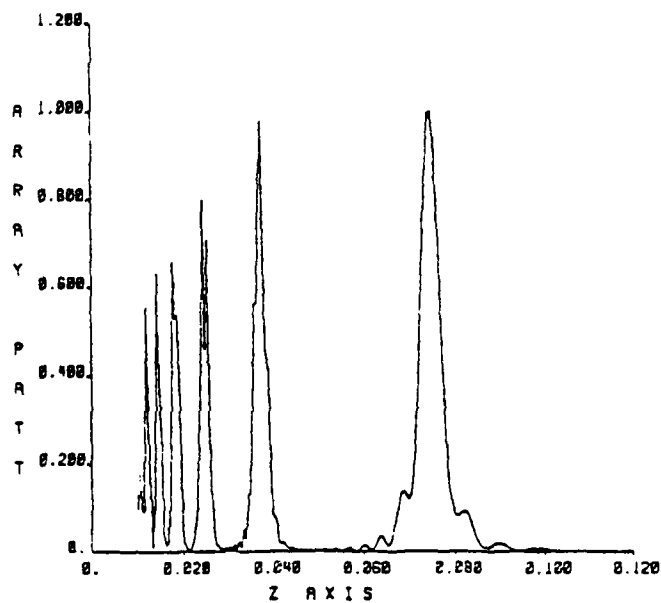


Figure 4.6.2 Fresnel Lens Array Axial Beam Pattern Showing False Focal Spots

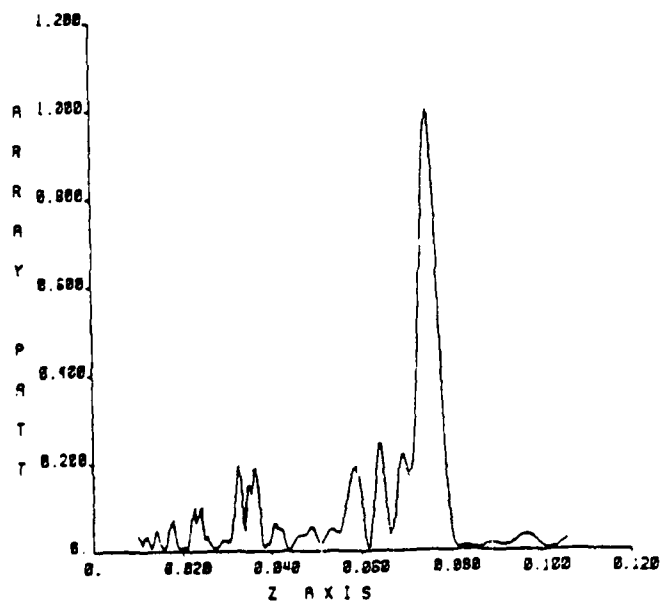


Figure 4.6.3 Fresnel Lens Array Axial Beam Pattern After Optimization

is shown in Figure 4.6.3. Again the procedure outlined by Madsen has proved invaluable.

For the purposes of this discussion attention is limited to suppressing false focal spots along the zone plate's axis, although joint sidelobe suppression in the principal focal plane is feasible.

Consider an electroded ring generating ultrasound $U(x_1, y_1)$ in an annulus with inner radius r and outer radius R .

$$U(x_1, y_1) = \begin{cases} a & r \leq \sqrt{x_1^2 + y_1^2} \leq R \\ 0 & \text{otherwise} \end{cases} \quad (4.6.3)$$

The diffracted ultrasound at a point $x_0 y_0$ in an observation plane a distance z from the plane of the annulus is approximately given by the Fresnel integral [78]

$$U(x_0, y_0; z) = \frac{\exp(jkz)}{j\lambda z} \exp \left\{ \frac{jk}{2z} (x_0^2 + y_0^2) \right\} \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U(x_1, y_1) \exp \left[\frac{jk}{2z} (x_1^2 + y_1^2) \right] \\ \exp \left\{ -j \frac{2\pi}{\lambda z} (x_0 x_1 + y_0 y_1) \right\} dx_1 dy_1 \quad (4.6.4)$$

where $k = \omega/c$.

For a sequence of N rings driven with amplitude a_n and pulse θ_n it has been shown [76] that

$$U(0,0;z) = \exp(jkz) \sum_{n=1}^N a_n e^{j\theta_n} \left[\exp\left(\frac{jk r_n^2}{2z}\right) - \exp\left(\frac{jk R_n^2}{2z}\right) \right] \quad (4.6.5)$$

where r_n and R_n are the inner and outer radii of each annulus and setting $y_0 = y_0 = 0$ restricts $U(\cdot)$ to axial observations.

In order to adjust r_n , R_n , a_n and θ_n to suppress the false focal spots using the Madsen routines, the problem is recast in the form of minimizing the maximum error, $F(\underline{V})$ where

$$F(\underline{V}) = \max_z |U(0,0;z) - U_D(z)| \quad (4.6.6)$$

over the z axis by adjusting the vector of parameters

$$\underline{V} = \{r_n, R_n, a_n, \theta_n ; n = 1, 2, \dots, N\} \quad (4.6.7)$$

The explicit derivative of $F(\underline{V})$ with respect to the \underline{V} variable similar to those of equations (4.5.2) through (4.5.9) have been calculated [76]. These were used to obtain results in the optimization of the axial beam pattern.

Two difficulties occur in this procedure. First, due to the complexity of the objective function, there are many local extrema which must be examined to arrive at the true global optimum. Second, optimization of even a sparse array tends to require that the spacing and thickness of at least a few of the rings be quite small. Failure to achieve such isolation will result in the ineffectiveness of the most tightly spaced rings resulting in substantial changes in the diffraction pattern.

The rings of the lenses were simulated by rings of many small individual elements. This allowed the use of existing programs for the previous array equations and derivatives and has been shown to give highly reliable results [76].

A second separate optimization was also performed on an 11 element Fresnel lens to reduce the "radial" grating lobes (or cones) inherent in these lenses. A cross-section of the array pattern similar to Figure 4.6.1 for the unoptimized array is shown in Figure 4.6.4. Note the very large side lobes containing considerably more energy than the main beam. After optimizing the gains and spacings as well as the ring thickness, Figure 4.6.5 was obtained. This is an improvement to the point where the array could be used successfully for imaging, which was not the case previously. Figures 4.6.6 and 4.6.7 show the two dimensional beam patterns of the unoptimized and optimized arrays respectively.

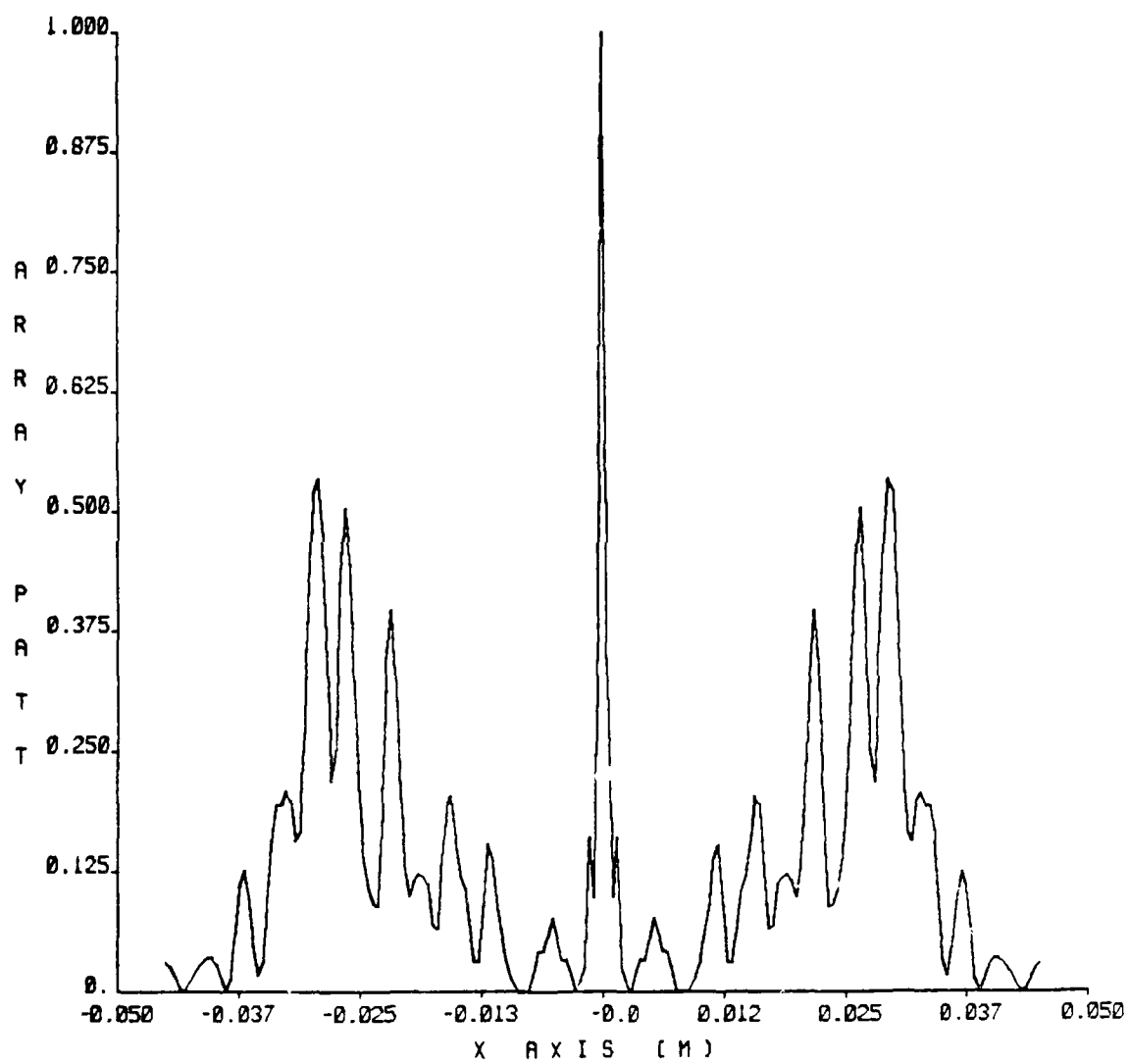


Figure 4.6.4 Cross Section of the Beam Pattern at the Focal Plane of an 11 Element Fresnel Lens Array

AD-A138 142

EMPLOYMENT OF ADAPTIVE LEARNING TECHNIQUES FOR THE
DISCRIMINATION OF ACOU. (U) GENERAL ELECTRIC CORPORATE
RESEARCH AND DEVELOPMENT SCHENECTA. J W ERKES ET AL.

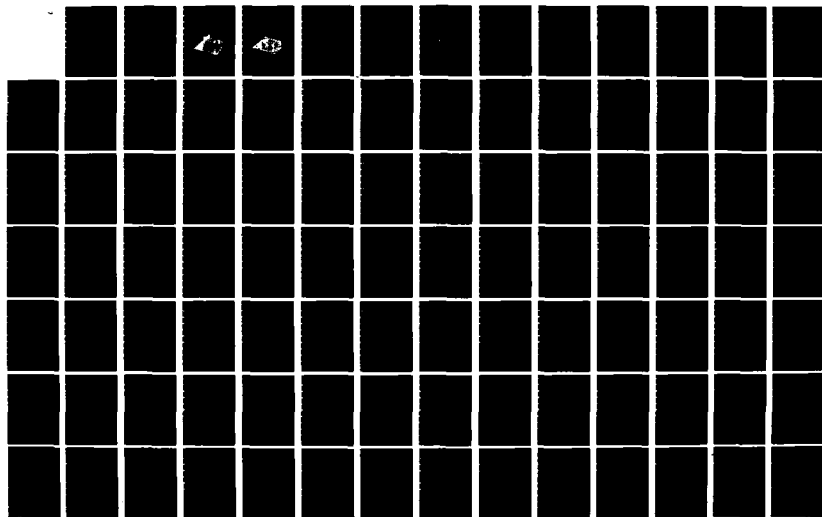
3/4

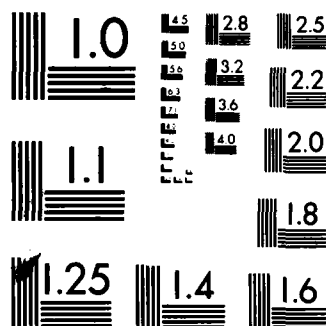
UNCLASSIFIED

NOV 83 83-SRD-060 N00014-82-C-2031

F/G 20/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

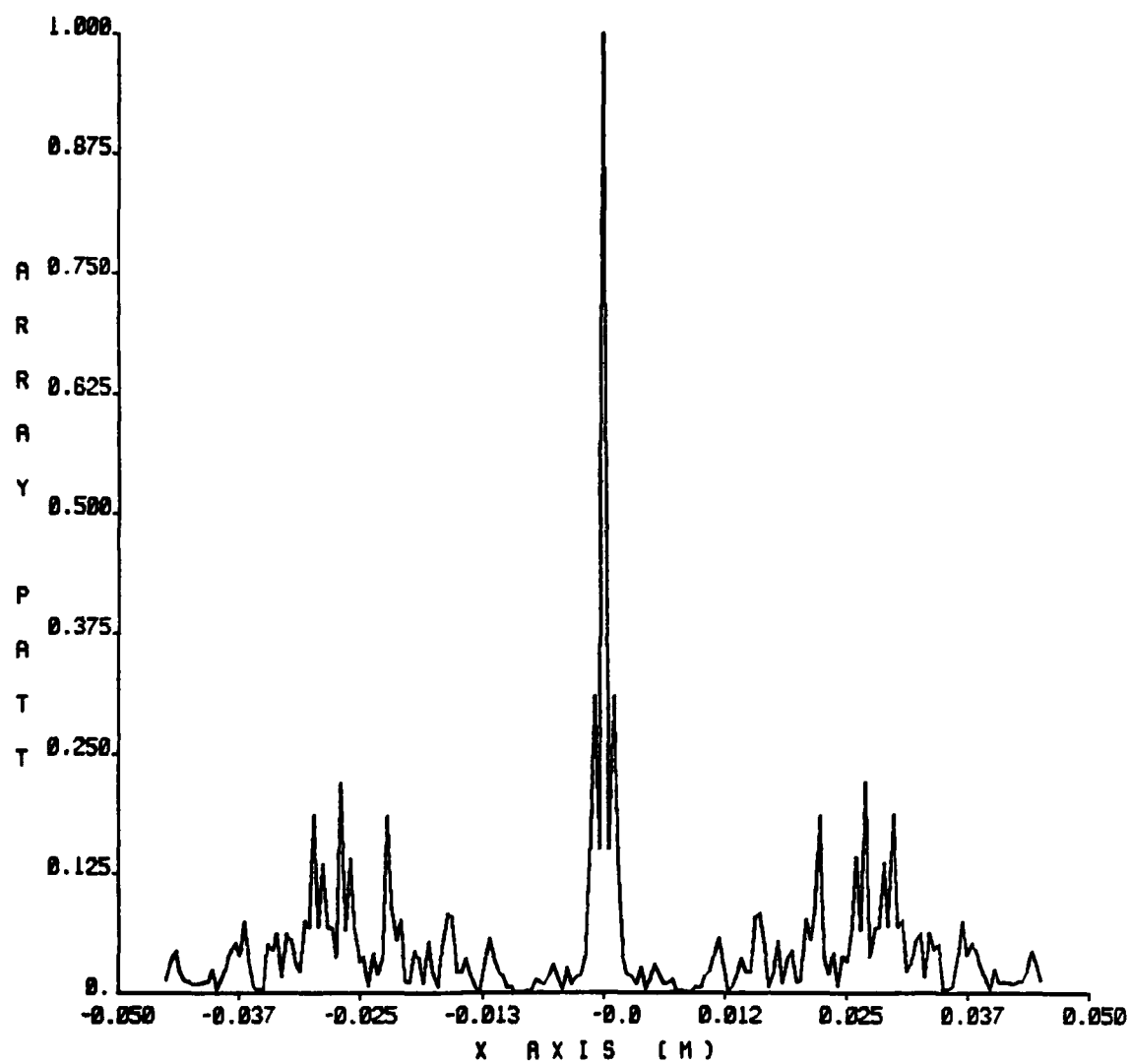


Figure 4.6.5 Cross Section of the Beam Pattern at the Focal Plane of an Optimized 11 Element Fresnel Lens Array

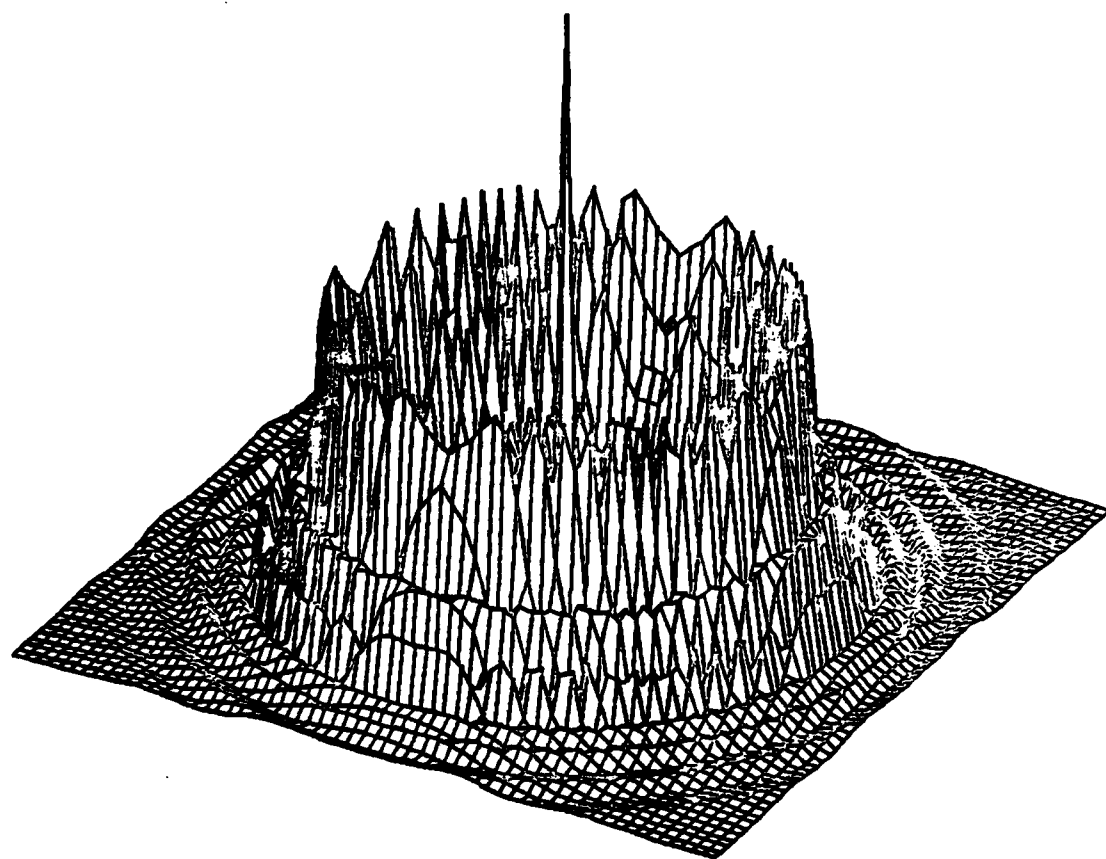


Figure 4.6.6 Beam Pattern of an 11 Element Fresnel Lens Array

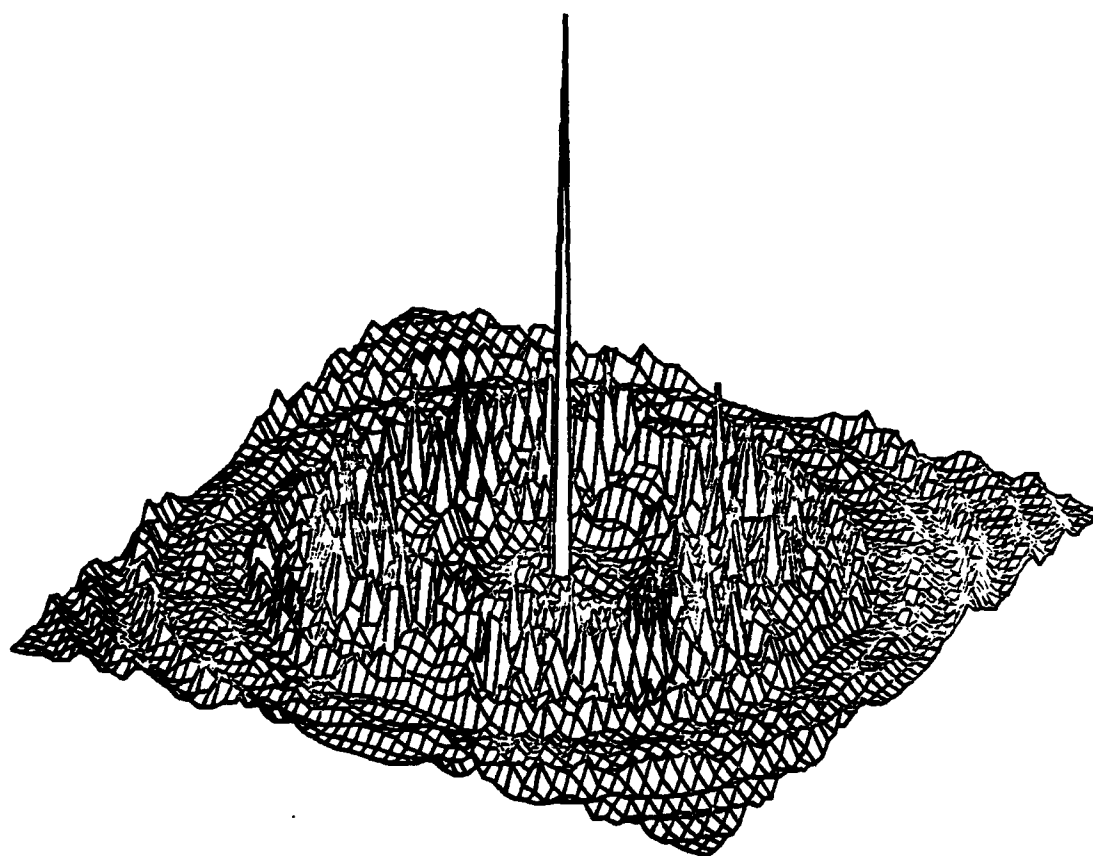


Figure 4.6.7 Beam Pattern of an Optimized 11 Element Fresnel Lens Array

4.7 Minimax Array Optimization Observations

It is important to note that due to the oscillatory nature of the objective function of Eq. (4.5.1) multiple extrema arise and it becomes necessary to examine them all to obtain the true global optimum. This can be a very costly and time-consuming computer search. In many cases results can be improved by slightly changing the initial array configuration and initial adjustment step size.

The Fresnel lens arrays are extremely sensitive to small changes in the ring spacings and any optimized design would have to be fabricated with extremely high precision. To build a working optimized array for ultrasonic applications would almost assuredly require the performance of trapped energy resonators [61,63,74,75,59], because of their high interelement isolation, small attainable element sizes and good bandwidths.

It has been shown here that antenna arrays can be optimized successfully for several different performances by adjusting the element gains, spacings and, for the Fresnel lens array, size. Although done separately here, different criteria may be optimized simultaneously but with the cost of an increased computational requirement. For example, one could simultaneously suppress grating lobes along the axis and grating rings in the focal plane. Further optimization may be done by removing the Cartesian constraint on Cartesian arrays and letting the elements reside anywhere in the plane. Much work still remains in speeding up the optimization

process and in establishing a result as being the true global optimum.

Tying the array optimization results to the dereverberation application produces a great increase in the quality of the source location images. Figure 4.7.1 demonstrates the improvement over Figure 3.6.8 when imaging is done with an optimized array.

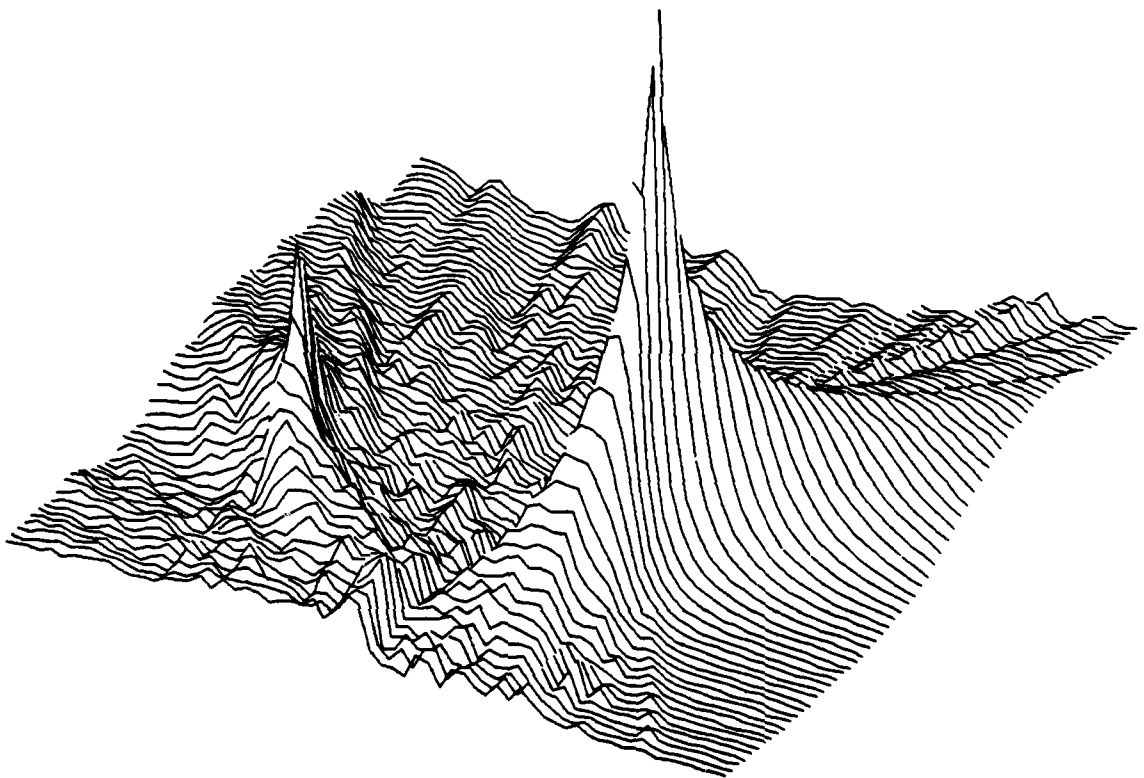


Figure 4.7.1 Optimized 11 Element Array Dereverberation Simulations

PART 5

DISCUSSION AND CONCLUSIONS

This work has presented several problems dealing with the many and varied applications of array antennas and has developed reasonable solutions for those problems which have been backed up with simulations. Each problem has been analyzed and fully understood before any approach was made towards a solution to insure the feasibility and usefulness of the given solution.

Dereverberation of the received signal particularly in non-destructive testing and flaw imaging was the first problem approached for both single source and dual source mediums. The single source solution was derived with some simplifying assumptions and rederived with added complications as the assumptions were removed. Finally a discussion was made on the conditions under which the theory breaks down altogether. Simulations showed some very promising results. The dual source case proved to be a problem several orders of magnitude more difficult than the single source case but with meticulous prodding, resulted in a solution with few limiting assumptions. Again simulations bore out the fruits of the labor of derivations.

No thesis is complete without mention of the known problems associated with a given method or topics for further investigation. The first to be noted is that the discrete effects of the DFT could be analyzed further since these effects must be carefully watched in all simulations. There are also problems with the method

when non-Hermite pulse shapes are being observed since perfect separation of signal and channel can no longer be achieved in this case. Furthermore, the relationship of the pulse shapes sent over each path from the source to the receiver (involving different modes and reflections for each path) is assumed constant from event to event and thereby can be described by a single transfer function which remains constant from event to event. Situations where the source spacial radiation pattern can change between events will violate this assumption in that the transfer function would change from event to event. Another problem area which is deserving of more attention is the computation of the cepstrum for signals whose spectrum is zero in a given band which would cause the log function to diverge. One possible simple solution is an analysis of the processing with the exclusion of such bands of frequencies. Finally one must be cautious of estimation noise in this processing since this will adversely affect the phase unwrapping results. This is an area in which much research could and should be done.

The array optimization problem which is more directly applicable to active transmit-receive imaging systems such as radar, ultrasonic and sonar was a somewhat smaller investigation. The question could be easily well posed, however the function was far from well behaved. Yet again with careful understanding and study some very good solutions were obtained with near optimal values of the elemental gains, positions and phases of one dimensional, two dimensional and fresnel lens arrays. Simulations once again demonstrated the validity of the solutions.

PART 6

LITERATURE CITED

1. Kraft, R.P., and J.F. McDonald, "Grating Lobe Suppression in a Correlation Acquisition Based Phased Array Imaging System," RPI Technical Report UI79-1, March 1979.
2. Oppenheim, A.V., and R.W. Schaeffer, Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1975.
3. Cramer, H., Mathematical Methods of Statistics, Princeton University Press, Princeton, NJ, 1963.
4. Kendall, M.G., and A. Stewart, The Advanced Theory of Statistics, Vol. 1, Hafner Publishing Co., NY, 1963.
5. Johnson, N.L., and S. Kotz, Distributions in Statistics: Continuous Multivariate Distributions, J. Wiley & Sons, NY, 1972.
6. Oppenheim, A.V., R.W. Schafer, and T.G. Stockham, "Nonlinear Filtering of Multiplied and Convolved Signals," Proc. of the IEEE, Vol. 56, No. 8, pp. 1264-1291, August 1968.
7. Tribolet, J.M., "A New Phase Unwrapping Algorithm," IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. ASSP-25, No. 2, April 1977.
8. Weinstein, C.J., J.W. Cooley, et al., Programs for Digital Signal Processing, IEEE Press, 1979 (Digital Signal Processing Committee, Ed.).
9. Tikhonov, V.I., and A.A. Tolkachev, "The Effects of Non-normal Fluctuations on Linear Systems," Nonlinear Transformations of Stochastic Processes, Pergamon Press, pp. 64-67, (P. I. Kuznetsov, et.al. Ed.), 1960.
10. Atal, B.S., "Linear Prediction for Speaker Identification," J. Acoust. Soc. Amer., Vol. 55, pp. 1304-1312, 1974.
11. Schroeder, M.R., "Direct (Nonrecursive) Relations Between Cepstrum and Predictor Coefficients," IEEE Trans. on Acoust., Speech, and Sig. Proc., Vol. ASSP-29, No. 2, pp. 297-301, April, 1981.
12. Papoulis, A., Probability, Random Variables, and Stochastic Processes, McGraw-Hill, NY, pp. 372-374, 1965.

13. Flanagan, J.L., Speech Analysis, Synthesis and Perception, Academic Press, New York, 1965.
14. Markel, J.D., A.H. Gray, Jr., Linear Prediction of Speech, Springer-Verlag, New York, 1976.
15. Bogert, B.P., M.S. Healy and J.W. Jukey, "The Frequency Analysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum, and Saphe Cracking," in M. Rosenblatt (ed.) Time Series Analysis, John Wiley & Sons, Inc., New York, Chap. 15, pp. 209-243, 1963.
16. Oppenheim, A.V., and R.W. Schaffer, "Homomorphic Analysis of Speech," IEEE Trans. Audio Electroacoust., Vol. Au-16, No. 2, pp. 221-226, June 1968.
17. Tribolet, J.M., T. F. Quatieri, "Computation of the Complex Cepstrum," Progress for Digital Signal Processing, (Digital Signal Processing Committee, Ed.), IEEE Press, 1979.
18. Quatieri, T.F., J.M. Tribolet, "Computation of the Real Cepstrum and Minimum-Phase Reconstruction," Programs for Digital Signal Processing, (Digital Signal Processing Committee, Ed.), IEEE Press, 1979.
19. Randall, R.B., and H. Hee, "Cepstrum Analysis," Brüel & Kjaer Instruments' Technical Review, Mass., 1981.
20. Tribolet, J.M., Seismic Applications of Homomorphic Signal Processing, Prentice-Hall, New Jersey, 1979.
21. Shapiro, A., "Approximation to the Array Signal-Gain Fluctuation Spectrum," J. Acoust. Soc. Am., Vol. 37, No. 6, pp. 1075-1078, June 1965.
22. Nuttall, A.H., and B.F. Cron, "Signal-Waveform Distortion Caused by Reflection off Lossy-Layered Bottoms," J. Acoust. Soc. Am., Vol. 40, No. 5, pp. 1094-1107, March 1966.
23. McClellan, J.H., T.W. Parks and L.R. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters," IEEE Trans. Audio Electroacoustics, Vol. AU-21, pp. 506-526, Dec. 1973.
24. Rabiner, L.R., J.H. McClellan, T.W. Parks, "FIR Digital Filter Design Techniques Using Weighted Chebyshev Approximation," Proc. IEEE, Vol. 63, pp. 592-610, April 1975.
25. Harris, F.J., "On the Use of Windows for Harmonic Analysis With the Discrete Fourier Transform," Proc. IEEE, Vol. 66, No. 1, pp. 51-83, Jan. 1978.

26. Kamp, Y., J.P. Thiran, "Chebyshev Approximation, for Two-Dimensional Nonrecursive Digital Filters," IEEE Trans. Circuits Syst., Vol. CAS-22, No. 3, pp. 208-218, March 1975.
27. Maeda, N., "Transversal Filters with Nonuniform Tap Spacings," IEEE Trans. Circuits Syst., Vol. CAS-27, No. 1, pp. 1-11, Jan. 1980.
28. Widrow, B., P.E. Mantey, L.J. Griffith, B.B. Goode, "Adaptive Antenna Systems," Proc. IEEE, Vol. 55, No. 12, pp. 2143-2159, Dec. 1967.
29. Griffiths, L.J., "A Simple Adaptive Algorithm for Real-Time Processing in Antenna Arrays," Proc. IEEE, Vol. 56, No. 10, pp. 1696-1704, Oct. 1969.
30. Mailloux, R.J., L. Zahn, A. Martinez, G.R. Forbes, "Grating Lobe Control in Limited Scan Arrays," IEEE Trans. Antennas Propagat., Vol. AP-27, No. 1, pp. 79-92, Jan. 1979.
31. Streit, R., "Sufficient Conditions for the Existence of Optimum Beam Patterns for Unequally Spaced Linear Arrays with an Example," IEEE Trans. Antennas Propagat., Vol. AP-23, No. 1, pp. 112-115, Jan. 1975.
32. Redlich, R., "Iterative Least-Squares Synthesis of Nonuniformly Spaced Linear Arrays," IEEE Trans. Antennas Propagat., Vol. AP-21, No. 1, pp. 106-108, Jan. 1973.
33. Wilson, G.L., "Computer Optimization of Transducer-Array Pattern," J. Acoust. Soc. Am., Vol. 59, No. 1, pp. 195-203, Jan. 1976.
34. Wilson, G.L., "The Design of Antenna Arrays with Tapered Sidelobe Heights," IEEE Trans. Antennas Propagat., Vol. AP-26, No. 2, pp. 345-347, March 1978.
35. Beaver, W.L., D.H. Doneron, A. Macowski, "Ultrasonic Imaging with an Acoustic Lens," IEEE Trans. Sonics Ultrasonics, Vol. SU-24, No. 4, pp. 235-243, July 1977.
36. Alais, P., B. Richard, "Progress in Fresnel Imaging; Clinical Evaluation," Acoustical Holography, Vol. 8, pp. 137-148, 1977.
27. Fink, M., "Theoretical Aspects of the Fresnel Focusing Technique," Acoustical Holography, Vol. 8, pp. 149-164, 1977.

38. Schjaer-Jacobsen, H., K. Madsen, "Synthesis of Nonuniformly Spaced Arrays Using a General Nonlinear Minimax Optimization Method," IEEE Trans. Antennas Propagat., Vol. AP-24, pp. 501-506, July 1976.
39. Madsen, K., "Minimal Solution of Non-Linear Equations Without Calculating Derivatives," Mathematical Programming Study 3, North-Holland, pp. 110-126, 1975.
40. King, D.D., R.F. Packard, R.K. Thomas, "Unequally Spaced, Broad Band Antenna Arrays," IRE Trans. on Antennas and Propagation, pp. 889-884, July 1960.
41. Madsen, K., "An Algorithm for Minimax Solution of Over Determined Systems of Non-Linear Equation," J. Inst. Maths. Applic., Vol. 16, pp. 321-328, December 1975.
42. Kraft, R.P., J.F. McDonald, F. Ahlgren, "Minimax Optimization of Two-Dimensional Focused Nonuniformly Spaced Arrays," Proc. 1979 IEEE Int. Conf. on Acoust, Speech and Signal Processing, pp. 286-289, April 1979.
43. Broyden, C.G., "A Class of Methods for Solving Nonlinear Simultaneous Equations," Math. Comp., Vol. 19, pp. 577-593, 1965.
44. Chien, M.J., "Searching for Multiple Solutions of Nonlinear Systems," IEEE Trans. Circuits Syst., Vol. CAS-26, No. 10, pp. 817-827.
45. Ecker, J.G. and M. Kupferschmid, "An Ellipsoid Algorithm for Nonlinear Programming," Mathematical Programming (to appear).
46. Smith, R.W., C.K. Day, "High Temperature Ultrasonic Transducers for In-Sodium Service," Technical Report HEDL-TME 75-21, Westinghouse Hanford Division, Richland, Washington 99352, January 1975.
47. Hoitnik, N.C., C. K. Day, "Under Sodium Viewing System Development for FFTF," Technical Report HEDL-TME 75-103, Westinghouse Hanford Division, Richland, Washington 99352, January 1975.
48. Sherman, J.W., "Properties of Focused Apertures in the Fresnel Region," IEEE Trans. on Antennas and Propagation, AP, July 1962, pp. 399-408.
49. Weeks, W.I., Antenna Engineering, McGraw-Hill, 1968.
50. Khurgin, Y.T., V.F. Yakovlev, "Progress in the Soviet Union on the Theory and Applications of Band Limited Functions," IEEE Proceedings, Vol. 64, No. 7, July 1977, pp. 1005-1029. [contains an interesting set of references].

51. Morse, P.M., K.U. Ingard, Theoretical Acoustics, McGraw-Hill, NY, 1968, p. 382.
52. Goodman, J.W., Introduction to Fourier Optics, McGraw-Hill, 1968.
53. Born, M., E. Wolf, Principles of Optics, Pergamon Press, New York, NY, 1959.
54. Evans, J.E., "Synthesis of Equiripple Sector Antenna Patterns," IEEE Trans. on Antennas and Propagation, May 1976.
55. Rabiner, L.R., B. Gold, Theory and Application of Digital Signal Processing, Prentice Hall, 1975.
56. Hamming, R.W., Digital Filters, Prentice Hall, 1977.
57. Rabiner, L.R., C.M. Rader, Digital Signal Processing, IEEE Press.
58. Dudgeon, D.E., "Fundamentals of Digital Array Processing," Proc. IEEE, Vol. 65, June 1977.
59. Amitey, N., V. Galindo, Chen Pang Wu, Theory and Analysis of Phased Array Antennas, Wiley Interscience.
60. Remez, E.Y., "General Computational Methods of Chebyshev Approximation," Keiv, 1957 (Atomic Energy Translation), 4491, pp. 1-85.
61. Tiersten, H.F., J.F. McDonald, M.F. Tse, P. Das, "Monolithic Mosaic Transducer Utilizing Trapped Energy Modes," Acoustical Holography, Vol. 7, L.W. Kessler (Ed.), pp. 405-421, 1977.
62. McClellan, J.H., "The Design of Two Dimensional Filters by Transformations," Proc. 7th Annual Princeton Conference on Information Sciences and Systems, pp. 247-251, 1973.
63. Tiersten, H.F., J.F. McDonald, P.K. Das, "Two Dimensional Monolithic Mosaic Transducer Array," Proc. IEEE Ultrasonics Symposium, pp. 406-412, 1977.
64. Steinberg, B.D., "Principles of Aperture and Array Systems Design," Wiley Interscience, 1976.
65. DuFort, E.C., "Constrained Feeds for Limited Scan Arrays," IEEE Trans. and Antennas and Propagation, Vol. AP-26, No. 3, pp. 407-413, May 1978.
66. Hodjat, F., S.A. Hovanessian, "Nonuniformly Spaced Linear and Planar Array Antennas for Side-lobe Reduction," IEEE Trans. on Antennas and Propagation, Vol. AP-26, No. 2, pp. 196-204, March 1978.

67. Harrington, R.F., "Sidelobe Reduction by Nonuniform Element Spacing," IRE Trans. on Antennas and Propagation, Vol. 9, No. 2, March 1961.
68. Thurstone, F.L., E.B. Miller, "Linear Ultrasonic Array for Echo-sonography," J. Acoust. Soc. Am., 61, pp. 1481-1491, June 1977.
69. Hildebrand, B.P., "An Analysis of Pulsed Ultrasonic Arrays," Proc. of the Eighth Acoustical Biography Symposium, 1978 (in press).
70. Harris, D.B., R.M. Mersereau, "A Comparison of Algorithms for Minimax Design of Two-Dimensional Linear Phase FIR Digital Filters," IEEE Trans. on Acoust., Speech and Sig. Process., ASSP-25 (#6), pp. 492-500, Dec. 1977.
71. Klein, M.V., Optics, J. Wiley & Sons, New York, NY, Chapter 8,
72. Farnow, S.A., B.A. Auld, "An Acoustic Phase Plate Imaging Device," in Acoustical Holography, Vol. 6, N. Booth (ed.), pp. 259-273, 1975.
73. Vilkomerson, D., B. Hurley, "Progress in Annular-Array Imaging," in Acoustical Holography, Vol. 6, N. Booth (ed.), pp. 145-164, 1975.
74. Tiersten, H.F., B.K. Sinha, J.F. McDonald, P.K. Das, "On the Influence of a Tuning Inductor on the Bandwidth of Extensional Trapped Energy Mode Transducers," Proc. 1978 IEEE Ultrasonics Symposium, pp. 163-166, 1978.
75. Das, P.K., G.A. White, B.K. Sinha, C. Lanzl, H.F. Tiersten, J.F. McDonald, "Ultrasonic Imaging Using Monolithic Mosaic Transducer Utilizing Trapped Energy Modes," in Acoustical Imaging, Vol. 8, A.F. Metherell (ed.), pp. 119-135, 1978.
76. Das, P.K., S. Talley, R. Kraft, H. Tiersten, J.F. McDonald, "Ultrasonic Imaging Using Trapped Energy Mode Fresnel Lens Transducers," Acoustical Imaging, Vol. 9, K.Y. Wang (ed.), pp. 75-92, 1980.
77. Wild, J.P., "A New Method of Image Formation with Annular Apertures and an Application to Radio Astronomy," Proc. Royal Society 286 A, 499, 1965.
78. Vilkomerson, D., "Acoustic Imaging with Thin Annular Apertures," In Acoustical Holography, Vol. 5, P.S. Green (ed.), Plenum Press, pp. 283-316, 1973.

79. Burkhardt, C.B., P.A. Grandchamp, H. Hoffman, "Methods for Increasing the Lateral Resolution of B-Scan," in Acoustical Holography, P.S. Green (ed.), Vol. 5, Plenum Press, pp. 391-413, 1973.
80. McDonald, J.F., A.A. Papadopoulos, "Grating and Aperture Sidelobe Optimization of Phased Arrays," Part I (Unfocused, Unsteered Arrays), R.P.I. Technical Report V1 78-1, Rensselaer Polytechnic Institute, June 16, 1978.
81. Rabiner, L.R., R.W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, NJ, pp. 208-216, 1978.
82. Bendat, J.S., Principles and Applications of Random Noise Theory, J. Wiley & Sons, pp. 258-261, 1958.
83. Breckenridge, F.R., "Acoustic Emission Transducer Calibration by Means of the Seismic Surface Pulse," J. Acoustic Emission, Vol. 1, No. 2, pp. 87-94, Jan. 1982.
84. Breckenridge, F.R., M. Greenspan, "Surface-Wave Displacement: Absolute Measurement Using a Capacitive Transducer," J. Acoust. Soc. Am., Vol. 69, No. 4, pp. 1177-1185, April 1981.
85. Hau, N.N., F.R. Breckenridge, "Characterization and Calibration of Acoustic Emission Sensors," Inter. Confr. Acoustic Emission, Sept. 1979.
86. Breckenridge, F.R., C.E. Tschiegg, M. Greenspan, "Acoustic Emission: Some Applications of Lamb's Problem," J. Acoust. Soc. Am., Vol. 56, No. 3, pp. 626-631, March 1975.

PART 7

APPENDICES

7.1 $F^{-1}[(j\omega)^i/i!]$ Simulation Program

(Subroutines from IEEE Programs for Digital Signal Processing)

C FTN DFFTJW

```
C     FTN DFFTJW
C     LOAD
C     LO B_DFFTJW
C     LI
C     SA *DFFTJW
C     Q
C     DELETE B_DFFTJW
C     O D.DFFTJW 1 2
C     R *DFFTJW
C     C 1
C     SEC #PLOT2D
C     D.DFFTJW
C     CO TTY
C     DIMENSION X(2048),Y(2048),R(2048),TH(2048),R1(2048),R2(2048)
C     K=3
C     LOG2N=11
C     N=2**LOG2N
C     SIGN=1.
C     ND2=N/2
C     XMIN=0.
C     XMAX=FLOAT(N)
C     M=ND2+1
C     DO 3 J=1,K
C     IPWR=(J-1)*4+1
C     II=MOD(IPWR,4)+1
C     GO TO (10,11,12,13),II
10  XI=1.
C     YI=0.
C     GO TO 14
11  XI=0.
C     YI=1.
C     GO TO 14
12  XI=-1.
C     YI=0.
C     GO TO 14
13  XI=0.
C     YI=-1.
14  CONTINUE
C     DO 1 I=1,ND2
C     W=(I-1.)/FLOAT(N)
C     X(I)=XI*W**IPWR/FACT(IPWR)
1    Y(I)=YI*W**IPWR/FACT(IPWR)
C     DO 2 I=M,N
C     W=(I-N-1.)/FLOAT(N)
C     X(I)=XI*W**IPWR/FACT(IPWR)
2    Y(I)=YI*W**IPWR/FACT(IPWR)
C    TAKE FFT OF SIGNAL
C    CALL MR1DF(LOG2N,X,Y,SIGN)
C    IF(J.EQ.1)CALL RP(X,Y,R,TH,N,0)
C    IF(J.EQ.2)CALL RP(X,Y,R1,TH,N,0)
C    IF(J.EQ.3)CALL RP(X,Y,R2,TH,N,0)
3    CONTINUE
C    DO 4 I=1,N
C    R(I)=ALOG10(R(I))
C    R1(I)=ALOG10(R1(I))
4    R2(I)=ALOG10(R2(I))
C    WRITE(5,100)N,K,XMIN,XMAX
```

C FTM DFFTJW

```
100 FORMAT(2I4,2G15.8)
WRITE(5,200)
200 FORMAT(' (8E15.8) ')
WRITE(5,300)((R(I),R1(I),R2(I)),I=1,N)
300 FORMAT(8E15.8)
CALL EXIT
END
```

```
C
C
C
FUNCTION FACT(I)
FACT=1.
IF(I.LE.1) RETURN
DO 1 N=1,I
1 FACT=FACT*FLOAT(N)
RETURN
END
```

```
C
C
C
SUBROUTINE RP(X,Y,R,TH,N,ID)
REC - POLAR CONVERSION
ID NOT =1 ==> R->P
ID =1 ==> P->R
DIMENSION X(2048),Y(2048),R(2048),TH(2048)
IF(ID.EQ.1) GO TO 10
DO 5 I=1,N
R(I)=SQRT(X(I)*X(I)+Y(I)*Y(I))
TH(I)=0.
IF(Y(I).EQ.0..AND.X(I).EQ.0.) GO TO 5
TH(I)=ATAN2(Y(I),X(I))
5 CONTINUE
RETURN
10 CONTINUE
DO 15 I=1,N
X(I)=R(I)*COS(TH(I))
15 Y(I)=R(I)*SIN(TH(I))
RETURN
END
```

```
C
C
C
SUBROUTINE MR1DF(LOG2N,X,Y,SIGN)
FORTRAN VERSION
MIXED RADIX FOURIER TRANSFORM
INTEGER LOG2N
REAL X(2048),Y(2048)
DIMENSION S(13),U(13)
```

```
C
C
C
INTEGER J1,J2,J3,J4,JT,N,M4
REAL ARG,C1,C2,C3,S1,S2,S3,R1,R2,R3,R4,R5,R6,R7,R8,T
```

```
C
C
C
INTEGER A,B,C,D,E,F,G,H,I,J,K,L,M.
1 BS,CS,DS,ES,FS,GS,HS,IS,JS,KS,LS,MS.
2 AL,BL,CL,DL,EL,FL,CL,HL,IL,JL,KL,ML,S,U
EQUIVALENCE (BS,S(2)),(CS,S(3)),(DS,S(4)),(ES,S(5)),(FS,S(6)),
```

C FTN DFFTJW

```

1 (GS,S(7)),(HS,S(8)),(IS,S(9)),(JS,S(10)),(KS,S(11)),(LS,S(12)),
2 (MS,S(13)),(AL,U(1)),(BL,U(2)),(CL,U(3)),(DL,U(4)),(EL,U(5)),
3 (FL,U(6)),(GL,U(7)),(HL,U(8)),(IL,U(9)),(JL,U(10)),(KL,U(11)),
4 (LL,U(12)),(ML,U(13))

```

C

```

      N=2**LOG2N
      IF (SIGN) 8000,8000,8002
8000  DO 8001 I=1,N
8001  Y(I)=-Y(I)
8002  CONTINUE
      IF (LOG2N-1) 500,500,501
501   CONTINUE
      DO 400 K=2,LOG2N,2
      M=2** (LOG2N-K)
      M4=4*M
      DO 300 J=1,M
          ARG=6.28315*FLOAT(J-1)/FLOAT(M4)
          C1=COS(ARG)
          S1=SIN(ARG)
          C2=C1*C1-S1*S1
          S2=C1*S1+C1*S1
          C3=C2*C1-S2*S1
          S3=C2*S1+S2*C1
          DO 200 I=M4,N,M4
              J1=I+J-M4
              J2=J1+M
              J3=J2+M
              J4=J3+M
              R1=X(J1)+X(J3)
              R2=X(J1)-X(J3)
              R3=Y(J1)+Y(J3)
              R4=Y(J1)-Y(J3)
              R5=X(J2)+X(J4)
              R6=X(J2)-X(J4)
              R7=Y(J2)+Y(J4)
              R8=Y(J2)-Y(J4)
              X(J1)=R1+R5
              Y(J1)=R3+R7
              IF (ARG) 101,100,101
101   CONTINUE
              X(J3)=(R2+R8)*C1+(R4-R6)*S1
              Y(J3)=(R4-R6)*C1-(R2+R8)*S1
              X(J2)=(R1-R5)*C2+(R3-R7)*S2
              Y(J2)=(R3-R7)*C2-(R1-R5)*S2
              X(J4)=(R2-R8)*C3+(R4+R6)*S3
              Y(J4)=(R4+R6)*C3-(R2-R8)*S3
              GO TO 200
100   CONTINUE
              X(J3)=R2+R8
              Y(J3)=R4-R6
              X(J2)=R1-R5
              Y(J2)=R3-R7
              X(J4)=R2-R8
              Y(J4)=R4+R6
200   CONTINUE
300   CONTINUE
400   CONTINUE

```

C FTM DFFTJW

```
500 CONTINUE
    ITEST=LOG2N-(LOG2N/2*2)
    IF(ITEST) 701,700,701
701 CONTINUE
    DO 600 I=1,N,2
    R1=X(I)+X(I+1)
    R2=X(I)-X(I+1)
    R3=Y(I)+Y(I+1)
    R4=Y(I)-Y(I+1)
    X(I)=R1
    Y(I)=R3
    X(I+1)=R2
    Y(I+1)=R4
600 CONTINUE
700 CONTINUE
    MS=N/2
    ML=N
    DO 800 K=2,12
    J=14-K
    S(J)=1
    U(J)=S(J+1)
    IF(S(J+1)-1) 7701,7701,7700
    S(J)=S(J+1)/2
7700 CONTINUE
7701 CONTINUE
800 CONTINUE
    AL=BS
    JT=0
    DO 900 A=1,AL
    DO 900 B=A,BL,BS
    DO 900 C=B,CL,CS
    DO 900 D=C,DL,DS
    DO 900 E=D,EL,ES
    DO 900 F=E,FL,FS
    DO 900 G=F,GL,GS
    DO 900 H=G,HL,HS
    DO 900 I=H,IL,IS
    DO 900 J=I,JL,JS
    DO 900 K=J,KL,KS
    DO 900 L=K,LL,LS
    DO 900 M=L,ML,MS
    JT=JT+1
    IF(JT-M) 900,900,901
901 CONTINUE
    T=X(JT)
    X(JT)=X(M)
    X(M)=T
    T=Y(JT)
    Y(JT)=Y(M)
    Y(M)=T
900 CONTINUE
    RETURN
    END
```

7.2 Phase Unwrapping and Cepstrum Computation Program

C

C

```
REAL*8 X(2048),Y(2048),R(2048),TH(2048),RM
REAL*8 DEXP,DBLE,DLOG10
REAL*4 XX(2048),CX(2050),AUX(2050)
COMMON PI,TWOPI,THLINC,THLCON,NFFT,NPTS,NN,L,H,H1,DVTM2
LOGICAL ISSUC
LOG2N=11
N=2**LOG2N
PI=4.*ATAN(1.)
TWOPI=2.*PI
THLINC=1.5
THLCON=.5
NFFT=N
SIGN=-1.
ND2=N/2
M=ND2+1
```

C DEFINE TIME PULSE

```
CENT=FLOAT(M)+0.
DO 1 I=1,N
X(I)=DBLE(ABS(N/2.-ABS(CENT-I)))    /(2.*2.*2.)
X(I)=DEXP(-X(I))
XX(I)=SNGL(X(I))
```

1 Y(I)=0.D0

CC DO 2 I=1,N

CC X(I)=X(I)+.5*DEXP(-DBLE(ABS(N/2.-ABS(CENT+128.-I)))/8.))

CC 9 XX(I)=SNGL(X(I))

C

C OUTPUT TIME SIGNAL

CALL WRI(X,Y,N,6)

C TAKE FFT OF TIME SIGNAL

CALL MR1DF(LOG2N,X,Y,SIGN)

CALL RP(X,Y,R,TH,N,0)

C OUTPUT FREQUENCY DOMAIN

CALL WRI(R,TH,N,5)

C

C FIND CEPSTRUM OF TIME SIGNAL

CALL CCEPS(N,XX,ISNX,ISFX,ISSUC,CX,AUX)

IF(.NOT. ISSUC) STOP

WRITE(1,300) ISNX,ISFX

300 FORMAT(G20.14)

J=N+ISFX

DO 3 I=1,N

3 X(I-ISFX)=DBLE(CX(I))

J=-ISFX

IF(J.LE.0) GO TO 5

DO 4 I=1,J

4 X(I)=DBLE(CX(N+ISFX+I))

5 CONTINUE

C OUTPUT CEPSTRUM

CALL WRI(X,X,N,7)

GO TO 40

C

C

C TAKE LOG OF FREQUENCY DOMAIN

21 DO 20 I=1,N

IF(R(I).LE.2.D-110) GO TO 22

X(I)=DLOG10(R(I))

C

```
      GO TO 20
22  X(I)=X(I-1)
20  Y(I)=TH(I)
C   OUTPUT LOG OF FREQUENCY DOMAIN
      CALL WRI(X,Y,N,9)
C   INVERT CEPSTRUM
40  CONTINUE
      DO 30 I=1,N
        R(I)=10.D0**X(I)
30  TH(I)=0.D0
      CALL RP(X,Y,R,TH,N,1)
      CALL MR1DF(LOG2N,X,Y,-1.)
      CALL RP(X,Y,R,TH,N,0)
C   OUTPUT INVERSE CEPSTRUM
      CALL WRI(R,TH,N,8)
      CALL EXIT
      END
```

C

C

C

```
      SUBROUTINE WRI(X,Y,N,K)
      REAL*8 X(2048),Y(2048)
      REAL*4 XK(2048),YY(2048)
      DO 10 I=1,N
        XK(I)=SNGL(X(I))
10  YY(I)=SNGL(Y(I))
      J=2
      XMIN=0.
      XMAX=FLOAT(N)
      WRITE(K,300)N,J,XMIN,XMAX
300  FORMAT(2I4,2G15.8)
      WRITE(K,200)
200  FORMAT(' (8E15.8) ')
      WRITE(K,100)((XK(I),YY(I)),I=1,N)
100  FORMAT(8E15.8)
      RETURN
      END
```

C

C

```
      SUBROUTINE RP(X,Y,R,TH,N,ID)
C   REC - POLAR CONVERSION
C   ID NOT = 1 ==> R->P
C   ID = 1 ==> P->R
      REAL*8 X(2048),Y(2048),R(2048),TH(2048)
      REAL*8 DSQRT,DATAN2,DCOS,DSIN
      IF(ID.EQ.1)GO TO 10
      DO 5 I=1,N
        R(I)=DSQRT(X(I)*X(I)+Y(I)*Y(I))
        TH(I)=0.D0
        IF(Y(I).EQ.0.D0.AND.X(I).EQ.0.D0)GO TO 5
        TH(I)=DATAN2(Y(I),X(I))
5     CONTINUE
      RETURN
10  CONTINUE
      DO 15 I=1,N
        X(I)=R(I)*DCOS(TH(I))
15  Y(I)=R(I)*DSIN(TH(I))
```

C

RETURN
END

C

C

C

C

C

C

SUBROUTINE MR1DF(LOC2N,X,Y,SIGN)

FORTRAN VERSION

MIXED RADIX FOURIER TRANSFORM

INTEGER LOC2N

REAL*8 X(2048),Y(2048)

DIMENSION S(13),U(13)

C

INTEGER J1,J2,J3,J4,JT,N,M4

REAL*8 ARG,C1,C2,C3,S1,S2,S3,R1,R2,R3,R4,R5,R6,R7,R2,T

REAL*8 DCOS,DSIN

C

C

C

INTEGER A,B,C,D,E,F,G,H,I,J,K,L,M,

1 BS,CS,DS,ES,FS,GS,HS,IS,JS,KS,LS,MS,

2 AL,BL,CL,DL,EL,FL,GL,HL,IL,JL,KL,ML,S,U

EQUIVALENCE (BS,S(2)),(CS,S(3)),(DS,S(4)),(ES,S(5)),(FS,S(6)),

1 (GS,S(7)),(HS,S(8)),(IS,S(9)),(JS,S(10)),(KS,S(11)),(LS,S(12)),

2 (MS,S(13)),(AL,U(1)),(BL,U(2)),(CL,U(3)),(DL,U(4)),(EL,U(5)),

3 (FL,U(6)),(GL,U(7)),(HL,U(8)),(IL,U(9)),(JL,U(10)),(KL,U(11)),

4 (LL,U(12)),(ML,U(13))

C

N=2**LOG2N

IF (SIGN) 8000,8000,8002

8000

DO 8001 I=1,N

8001

Y(I)=-Y(I)

8002

CONTINUE

IF (LOG2N-1) 500,500,501

501

CONTINUE

DO 400 K=2,LOG2N,2

M=2**((LOG2N-K)

M4=4*M

DO 300 J=1,M

ARG=6.28315D0*DBLE(FLOAT(J-1)/FLOAT(M4))

C1=DCOS(ARG)

S1=DSIN(ARG)

C2=C1*C1-S1*S1

S2=C1*S1+C1*S1

C3=C2*C1-S2*S1

S3=C2*S1+S2*C1

DO 200 I=M4,N,M4

J1=I+J-M4

J2=J1+M

J3=J2+M

J4=J3+M

R1=X(J1)+X(J3)

R2=X(J1)-X(J3)

R3=Y(J1)+Y(J3)

R4=Y(J1)-Y(J3)

R5=X(J2)+X(J4)

R6=X(J2)-X(J4)

R7=Y(J2)+Y(J4)

C

```

R8=Y(J2)-Y(J4)
X(J1)=R1+R5
Y(J1)=R3+R7
IF(ARG) 101,100,101
101 CONTINUE
X(J3)=(R2+R8)*C1+(R4-R6)*S1
Y(J3)=(R4-R5)*C1-(R2+R8)*S1
X(J2)=(R1-R5)*C2+(R3-R7)*S2
Y(J2)=(R3-R7)*C2-(R1-R5)*S2
X(J4)=(R2-R8)*C3+(R4+R6)*S3
Y(J4)=(R4+R6)*C3-(R2-R8)*S3
GO TO 200
100 CONTINUE
X(J3)=R2+R8
Y(J3)=R4-R6
X(J2)=R1-R5
Y(J2)=R3-R7
X(J4)=R2-R8
Y(J4)=R4+R6
200 CONTINUE
300 CONTINUE
400 CONTINUE
500 CONTINUE
ITEST=LOG2N-(LOG2N/2*2)
IF(ITEST) 701,700,701
701 CONTINUE
DO 600 I=1,N,2
R1=X(I)+X(I+1)
R2=X(I)-X(I+1)
R3=Y(I)+Y(I+1)
R4=Y(I)-Y(I+1)
X(I)=R1
Y(I)=R3
X(I+1)=R2
Y(I+1)=R4
600 CONTINUE
700 CONTINUE
MS=N/2
ML=N
DO 800 K=2,12
J=14-K
S(J)=1
U(J)=S(J+1)
IF(S(J+1)-1) 7701,7701,7700
7700 S(J)=S(J+1)/2
7701 CONTINUE
800 CONTINUE
AL=BS
JT=0
DO 900 A=1,AL
DO 900 B=A,BL,BS
DO 900 C=B,CL,CS
DO 900 D=C,DL,DS
DO 900 E=D,EL,ES
DO 900 F=E,FL,FS
DO 900 G=F,GL,GS
DO 900 H=G,HL,HS

```

C

```
DO 900 I=H, IL, IS
DO 900 J=I, JL, JS
DO 900 K=J, KL, KS
DO 900 L=K, LL, LS
DO 900 M=L, ML, MS
JT=JT+1
```

```
          IF(JT-MD 900,900,901
901      CONTINUE
      T=X(JT)
      X(JT)=X(MD)
      X(MD)=T
      T=Y(JT)
      Y(JT)=Y(MD)
      Y(MD)=T
900 CONTINUE
      RETURN
      END
```

```

C
C
C   COMPILE:   FTM CCEPS -64V
C
C   LOAD:      LOAD B_CCEPS SUBROUTINES TO PROGS
C               USING CEPSTRUM CALCULATIONS
C-----
C SUBROUTINE: CCEPS
C SUBROUTINE TO COMPUTE THE COMPLEX CEPSTRUM OF A SEQUENCE X(N)
C-----
C
C   SUBROUTINE CCEPS(NX,X,ISNX,ISFX,ISSUC,CX,AUX)
C
C   DIMENSION X(1),CX(1),AUX(1)
C   COMMON P1,TWOP1,THLINC,THLCON,NFFT,NPTS,N,L,H,H1,DVTM2
C   LOGICAL ISSUC
C   NPTS=NFFT/2
C   N=12
C   L=2**N
C   H=FLOAT(L)*FLOAT(NFFT)
C   H1=PI/H
C   ISSUC=.TRUE.
C   ISNX=1
C
C   DO 10 I=1,NX
C     CX(I)=X(I)
C     AUX(I)=FLOAT(I-1)*X(I)
C 10 CONTINUE
C   INITL=NX+1
C   IEND=NFFT+2
C   DO 20 I=INITL,IEND
C     CX(I)=0.0
C     AUX(I)=0.0
C 20 CONTINUE
C
C   CALL FFA(CX,NFFT)
C   CALL FFA(AUX,NFFT)
C
C   IF(CX(1).LT.0.0) ISNX=-1
C
C   IO=-1
C   DVTM2=0.0
C   IEND=NPTS+1
C   DO 30 I=1,IEND
C     IO=IO+2
C     IE=IO+1
C     AMAGSQ=AMODSQ(CX(IO),CX(IE))
C     PDVT=PHADVT(CX(IO),CX(IE),AUX(IO),AUX(IE),AMAGSQ)
C     AUX(IO)=AMAGSQ
C     AUX(IE)=PDVT
C     DVTM2=DVTM2+PDVT
C 30 CONTINUE
C   DVTM2=(2.*DVTM2-AUX(2)-PDVT)/FLOAT(NPTS)
C
C   PPDVT=AUX(2)
C   PPHASE=0.0
C   PPV=PI*VPIA(CX(1),CX(2),ISNX)
C   CX(1)=.5*ALOG(AUX(1))
C   CX(2)=0.0
C   IO=1

```

```

C
DO 50 I=2,IEND
  IO=IO+2
  IE=IO+1
  PDVT=AUX(IE)
  PPV=PPVPHA(CX(IO),CX(IE),ISNX)
  PHASE=PHAUNW(X,NX,ISNX,I,PPHASE,PPDVT,PPV,PDVT,ISSUC)
C
  IF(ISSUC)GO TO 40
  ISSUC=.FALSE.
  RETURN
40  PPDVT=PDVT
  PPHASE=PHASE
  CX(IO)=.5*ALOG(AUX(IO))
  CX(IE)=PHASE
50  CONTINUE
C
  ISFX=(ABS(PHASE/PI)+.1)
  IF(PHASE.LT.0.0)ISFX=-ISFX
  H=PHASE/FLOAT(NPTS)
  IE=0
  DO 60 I=1,IEND
    IE=IE+2
    CX(IE)=CX(IE)-H*FLOAT(I-1)
60  CONTINUE
C
  CALL FFS(CX,NFFT)
  RETURN
  END
C
C-----
C SUBROUTINE: SPCVAL
C SUBROUTINE TO COMPUTRE A SPECTRAL VALUE AT A FREQUENCY
C FREQ(RADIANS) FOR SEQUENCE X(N) AND N*X(N)
C-----
C
SUBROUTINE SPCVAL(NX,X,FREQ,XR,XI,YR,YI)
  DIMENSION X(1)
  DOUBLE PRECISION U0,U1,U2,W0,W1,W2,A,B,C,D,A1,A2,SA0,CA0
C
  CA0=DBLE(COS(FREQ))
  SA0=DELE(SIN(FREQ))
  A1=2.D+0*CA0
  U1=0.D+0
  U2=U1
  W1=U1
  W2=U1
C
  DO 10 J=1,NX
    XJ=DBLE(X(J))
    U0=XJ+A1*U1-U2
    W0=DBLE(FLOAT(J-1))*XJ+A1*W1-W2
    U2=U1
    U1=U0
    W2=W1
    W1=W0
10  CONTINUE
C
  A=U1-U2*CA0

```

C

```
B=U2*SA0
C=W1-W2*CA0
D=W2*SA0
A2=DBLE(FREQ*FLOAT(NX-1))
U1=DCOS(A2)
U2=-DSIN(A2)
XR=SNGL(U1*A-U2*B)
XI=SNGL(U2*A+U1*B)
YR=SNGL(U1*C-U2*D)
YI=SNGL(U2*C+U1*D)
RETURN
END
```

C

C

```
C FUNCTION: PHAUNW
C PHASE UNWRAPPING BASED ON TRIBOLET'S ADAPTIVE INTEGRATION SCHEME.
C THE UNWRAPPED PHASE ESTIMATE IS RETURNED IN PHAUNW.
```

C

C

C

```
DIMENSION SDVT(17),SPPV(17),X(1)
INTEGER SINDEXT(17),PINDEXT,SP
LOGICAL ISCONS,FIRST
COMMON PI,TWOPI,THLINC,THLCON,NFFT,NPTS,N,L,H,H1,DVTM2
```

C

```
FIRST=.TRUE.
PINDEXT=1
SP=1
SPPV(SP)=PPV
SDVT(SP)=PDVT
SINDEXT(SP)=L+1
```

C

```
GO TO 40
```

C

```
10 PINDEXT=SINDEXT(SP)
PPHASE=PHASE
PPDVT=SDVT(SP)
SP=SP-1
GO TO 40
```

C

```
20 IF((SINDEXT(SP)-PINDEXT).GT.1)GO TO 30
ISCONS=.FALSE.
PHAUNW=0.
RETURN
```

C

```
30 K=(SINDEXT(SP)+PINDEXT)/2
```

C

```
FREQ=TWOPI*(FLOAT(I-2)*FLOAT(L)+FLOAT(K-1))/H
CALL SPCVAL(NX,X,FREQ,XR,XI,YR,YI)
```

C

```
SP=SP+1
SINDEXT(SP)=K
SPPV(SP)=PPVPHA(XR,XI,ISNX)
XMAC=AMODSQ(XR,XI)
SDVT(SP)=PHADVT(XR,XI,YR,YI,XMAC)
```

C

```
40 DELTA=H1*FLOAT(SINDEXT(SP)-PINDEXT)
PHAINC=DELTA*(PPDVT+SDVT(SP))
```



```

C
C
C      IF(ABS(PHAINC-DELTA*DVTM2).GT.THLINC)GO TO 20
C
C      PHASE=PPHASE+PHAINC
C      CALL PHCHCK(PHASE,SPPV(SP),ISCONS)
C      IF(.NOT.ISCONS)GO TO 20
C
C      IF(ABS(PHASE-PPHASE).GT.PI)GO TO 20
C
C      IF(SP.NE.1)GO TO 10
C      PHAUN/=PHASE
C      RETURN
C      END
C
C-----
C FUNCTION: PPVPHA
C COMPUTE THE PRINCIPLE VALUE OF THE PHASE OF A SPECTRAL VALUE
C-----
C
C      FUNCTION PPVPHA(XR,XI,ISNX)
C
C      IF(ISNX.EQ.1)PPVPHA=(ATAN2((XI),(XR)))
C      IF(ISNX.EQ.(-1))PPVPHA=(ATAN2(-(XI),-(XR)))
C      RETURN
C      END
C
C-----
C FUNCTION: PHADVT
C COMPUTE THE PHASE DERIVATIVE OF A SPECTRAL VALUE OF A SEQUENCE X(N)
C-----
C
C      FUNCTION PHADVT(XR,XI,YR,YI,XMAG)
C
C      PHADVT=-SNGL((DBLE(XR)*DBLE(YR)+DBLE(XI)*DBLE(YI))/DBLE(XMAG))
C      RETURN
C      END
C
C-----
C FUNCTION: AMODSQ
C COMPUTE THE SQUARE OF THE MODULUS OF A COMPLEX NUMBER
C-----
C
C      FUNCTION AMODSQ(ZR,ZI)
C
C      AMODSQ=SNGL(DBLE(ZR)*DBLE(ZR)+DBLE(ZI)*DBLE(ZI))
C      RETURN
C      END
C
C-----
C SUBROUTINE: PHCHCK
C SUBROUTINE TO CHECK CONSISTENCY OF A PHASE ESTIMATE
C-----
C
C      SUBROUTINE PHCHCK(PH,PV,ISCONS)
C
C      COMMON PI,TWOPI,THLINC,THLCON,NFFT,NPTS,N,L,H,H1,DVTM2
C      LOGICAL ISCONS
C
C      A0=(PH-PV)/TWOPI

```

C

```
A1=FLOAT(IFIX(A0))*TWOPI+PV
A2=A1+SIGN(TWOPI,A0)
A3=ABS(A1-PII)
A4=ABS(A2-PII)
```

C

```
ISCONS=.FALSE.
IF(A3.GT.THLCON.AND.A4.GT.THLCON)RETURN
ISCONS=.TRUE.
```

C

```
PII=A1
IF(A3.GT.A4)PII=A2
RETURN
END
```

C

C-----
C SUBROUTINE: FFA
C FAST FOURIER ANALYSIS SUBROUTINE
C-----

C

```
SUBROUTINE FFA(B, NFFT)
```

C

C THIS SUBROUTINE REPLACES THE REAL VECTOR B(K), (K=1,2,...,N),
C WITH ITS FINITE DISCRETE FOURIER TRANSFORM. THE DC TERM IS
C RETURNED IN LOCATION B(1) WITH B(2) SET TO 0. THEREAFTER, THE
C JTH HARMONIC IS RETURNED AS A COMPLEX NUMBER STORED AS
C B(2*J+1) + I B(2*J+2). NOTE THAT THE N/2 HARMONIC IS RETURNED
C IN B(N+1) WITH B(N+2) SET TO 0. HENCE, B MUST BE DIMENSIONED
C TO SIZE N+2.
C SUBROUTINE IS CALLED AS FFA (B,N) WHERE N=2**M AND B IS AN
C N TERM REAL ARRAY. A REAL-VALUED, RADIX 8 ALGORITHM IS USED
C WITH IN-PLACE REORDERING AND THE TRIG FUNCTIONS ARE COMPUTED AS
C NEEDED.

C

```
DIMENSION B(2)
COMMON /CON/ PII, P7, P7TWO, C22, S22, P12
```

C

C IW IS A MACHINE DEPENDENT WRITE DEVICE NUMBER

C

```
IW = IIMACH(2)
IW=1
```

C

```
PII = 4.*ATAN(1.)
P18 = PII/8.
P7 = 1./SQRT(2.)
P7TWO = 2.*P7
C22 = COS(P18)
S22 = SIN(P18)
P12 = 2.*PII
N = 1
DO 10 I=1,15
  M = I
  N = N*2
  IF (N.EQ.NFFT) GO TO 20
```

10 CONTINUE

```
WRITE (IW,9999)
```

9999 FORMAT (30H NFFT NOT A POWER OF 2 FOR FFA)

```
STOP
```

20 CONTINUE

```
NSPOW = M/3
```

C

C

C DO A RADIX 2 OR RADIX 4 ITERATION FIRST IF ONE IS REQUIRED

C

IF (M-NSPOW*3-1) 50, 40, 30

30 NN = 4

INT = N/NN

CALL R4TR(INT, B(1), B(INT+1), B(2*INT+1), B(3*INT+1))

GO TO 60

40 NN = 2

INT = N/NN

CALL R2TR(INT, B(1), B(INT+1))

GO TO 60

50 NN = 1

C

C PERFORM RADIX 8 ITERATIONS

C

60 IF (NSPOW) 90, 90, 70

70 DO 80 IT=1, NSPOW

NN = NN*8

INT = N/NN

CALL R8TR(INT, NN, B(1), B(INT+1), B(2*INT+1), B(3*INT+1),

* B(4*INT+1), B(5*INT+1), B(6*INT+1), B(7*INT+1), B(1),

* B(INT+1), B(2*INT+1), B(3*INT+1), B(4*INT+1), B(5*INT+1),

* B(6*INT+1), B(7*INT+1))

80 CONTINUE

C

C PERFORM IN-PLACE REORDERING

C

90 CALL ORD1(M, B)

CALL ORD2(M, B)

T = B(2)

B(2) = 0.

B(NFFT+1) = T

B(NFFT+2) = 0.

DO 100 I=4, NFFT, 2

B(I) = -B(I)

100 CONTINUE

RETURN

END

C

C

C SUBROUTINE: FFS

C FAST FOURIER SYNTHESIS SUBROUTINE

C RADIX 8-4-2

C

C

SUBROUTINE FFS(B, NFFT)

C

C THIS SUBROUTINE SYNTHESIZES THE REAL VECTOR B(K), WHERE

C K=1,2,...,N. THE INITIAL FOURIER COEFFICIENTS ARE PLACED IN

C THE B ARRAY OF SIZE N+2. THE DC TERM IS IN B(1) WITH

C B(2) EQUAL TO 0.

C THE JTH HARMONIC IS STORED AS B(2*J+1) + I B(2*J+2).

C THE N/2 HARMONIC IS IN B(N+1) WITH B(N+2) EQUAL TO 0.

C THE SUBROUTINE IS CALLED AS FFS(B,N) WHERE N=2**M AND

C B IS THE N TERM REAL ARRAY DISCUSSED ABOVE.

C

DIMENSION B(2)

COMMON /CON1/ P11, P7, P7TWO, C22, S22, P12

```

C
C
C 1W IS A MACHINE DEPENDENT WRITE DEVICE NUMBER
C
C      1W = 11MACH(2)
C      1W=1
C
      P11 = 4.*ATAN(1.)
      P18 = P11/8.
      P7 = 1./SQRT(2.)
      P7TWO = 2.*P7
      C22 = COS(P18)
      S22 = SIN(P18)
      P12 = 2.*P11
      N = 1
      DO 10 I=1,15
        M = I
        N = N*2
        IF (N.EQ.NFFT) GO TO 20
10    CONTINUE
      WRITE (1W,9999)
9999  FORMAT (30H NFFT NOT A POWER OF 2 FOR FFS)
      STOP
20    CONTINUE
      B(2) = B(NFFT+1)
      DO 30 I=1,NFFT
        B(I) = B(I)/FLOAT(NFFT)
30    CONTINUE
      DO 40 I=4,NFFT,2
        B(I) = -B(I)
40    CONTINUE
      N8POW = M/3
C
C REORDER THE INPUT FOURIER COEFFICIENTS
C
      CALL ORD2(M, B)
      CALL ORD1(M, B)
C
      IF (N8POW.EQ.0) GO TO 60
C
C PERFORM THE RADIX 8 ITERATIONS
C
      NN = N
      DO 50 IT=1,N8POW
        INT = N/NN
        CALL R8SYN(INT, NN, B, B(INT+1), B(2*INT+1), B(3*INT+1),
*          B(4*INT+1), B(5*INT+1), B(6*INT+1), B(7*INT+1), B(1),
*          B(INT+1), B(2*INT+1), B(3*INT+1), B(4*INT+1), B(5*INT+1),
*          B(6*INT+1), B(7*INT+1))
        NN = NN/8
50    CONTINUE
C
C DO A RADIX 2 OR RADIX 4 ITERATION IF ONE IS REQUIRED
C
60    IF (N-N8POW*3-1) 90, 80, 70
70    INT = N/4
      CALL R4SYN(INT, B(1), B(INT+1), B(2*INT+1), B(3*INT+1))
      GO TO 90
80    INT = N/2
      CALL R2TR(INT, B(1), B(INT+1))

```

C

90 RETURN
END

C

C

C SUBROUTINE: R2TR
C RADIX 2 ITERATION SUBROUTINE

C

C

C

SUBROUTINE R2TR(INT, B0, B1)
DIMENSION B0(2), B1(2)
DO 10 K=1,INT
T = B0(K) + B1(K)
B1(K) = B0(K) - B1(K)
B0(K) = T

10 CONTINUE
RETURN
END

C

C

C SUBROUTINE: R4TR
C RADIX 4 ITERATION SUBROUTINE

C

C

SUBROUTINE R4TR(INT, B0, B1, B2, B3)
DIMENSION B0(2), B1(2), B2(2), B3(2)
DO 10 K=1,INT
R0 = B0(K) + B2(K)
R1 = B1(K) + B3(K)
B2(K) = B0(K) - B2(K)
B3(K) = B1(K) - B3(K)
B0(K) = R0 + R1
B1(K) = R0 - R1

10 CONTINUE
RETURN
END

C

C

C SUBROUTINE: R8TR
C RADIX 8 ITERATION SUBROUTINE

C

C

SUBROUTINE R8TR(INT, NN, BR0, BR1, BR2, BR3, BR4, BR5, BR6, BR7,
* B10, B11, B12, B13, B14, B15, B16, B17)
DIMENSION L(15), BR0(2), BR1(2), BR2(2), BR3(2), BR4(2), BR5(2),
* BR6(2), BR7(2), B10(2), B11(2), B12(2), B13(2), B14(2),
* B15(2), B16(2), B17(2)
COMMON /CON/ P11, P7, P7TWO, C22, S22, P12
EQUIVALENCE (L15,L(1)), (L14,L(2)), (L13,L(3)), (L12,L(4)),
* (L11,L(5)), (L10,L(6)), (L9,L(7)), (L8,L(8)), (L7,L(9)),
* (L6,L(10)), (L5,L(11)), (L4,L(12)), (L3,L(13)), (L2,L(14)),
* (L1,L(15))

C

C SET UP COUNTERS SUCH THAT JTHET STEPS THROUGH THE ARGUMENTS
C OF W, JR STEPS THROUGH STARTING LOCATIONS FOR THE REAL PART OF THE
C INTERMEDIATE RESULTS AND JI STEPS THROUGH STARTING LOCATIONS
C OF THE IMAGINARY PART OF THE INTERMEDIATE RESULTS.

C

L(1) = NN/8

C

```

DO 40 K=2,15
  IF (L(K-1)-2) 10, 20, 30
10  L(K-1) = 2
20  L(K) = 2
   GO TO 40
30  L(K) = L(K-1)/2
40  CONTINUE
   PLOVN = P11/FLOAT(NN)
   J1 = 3
   JL = 2
   JR = 2
DO 120 J1=2,L1,2
DO 120 J2=J1,L2,L1
DO 120 J3=J2,L3,L2
DO 120 J4=J3,L4,L3
DO 120 J5=J4,L5,L4
DO 120 J6=J5,L6,L5
DO 120 J7=J6,L7,L6
DO 120 J8=J7,L8,L7
DO 120 J9=J8,L9,L8
DO 120 J10=J9,L10,L9
DO 120 J11=J10,L11,L10
DO 120 J12=J11,L12,L11
DO 120 J13=J12,L13,L12
DO 120 J14=J13,L14,L13
DO 120 JTHET=J14,L15,L14
   TH2 = JTHET - 2
   IF (TH2) 50, 50, 90
50  DO 60 K=1,INT
     T0 = BR0(K) + BR4(K)
     T1 = BR1(K) + BR5(K)
     T2 = BR2(K) + BR6(K)
     T3 = BR3(K) + BR7(K)
     T4 = BR0(K) - BR4(K)
     T5 = BR1(K) - BR5(K)
     T6 = BR2(K) - BR6(K)
     T7 = BR3(K) - BR7(K)
     BR2(K) = T0 - T2
     BR3(K) = T1 - T3
     T0 = T0 + T2
     T1 = T1 + T3
     BR0(K) = T0 + T1
     BR1(K) = T0 - T1
     PR = P7*(T5-T7)
     PI = P7*(T5+T7)
     BR4(K) = T4 + PR
     BR7(K) = T6 + PI
     BR6(K) = T4 - PR
     BR5(K) = PI - T6
60  CONTINUE
   IF (NN-8) 120, 120, 70
70  K0 = INT*8 + 1
   KL = K0 + INT - 1
DO 80 K=K0,KL
   PR = P7*(BI2(K)-BI6(K))
   PI = P7*(BI2(K)+BI6(K))
   TR0 = BI0(K) + PR
   TI0 = BI4(K) + PI
   TR2 = BI0(K) - PR

```

C

```

T12 = B14(K) - PI
PR = P7*(B13(K)-B17(K))
PI = P7*(B13(K)+B17(K))
TR1 = B11(K) + PR
TI1 = B15(K) + PI
TR3 = B11(K) - PR
TI3 = B15(K) - PI
PR = TR1*C22 - TI1*S22
PI = TI1*C22 + TR1*S22
BI0(K) = TR0 + PR
BI6(K) = TR0 - PR
BI7(K) = TI0 + PI
BI1(K) = PI - TI0
PR = -TR3*S22 - TI3*C22
PI = TR3*C22 - TI3*S22
BI2(K) = TR2 + PR
BI4(K) = TR2 - PR
BI5(K) = TI2 + PI
BI3(K) = PI - TI2

```

80

CONTINUE

GO TO 120

90

ARG = TH2*PIOVN

C1 = COS(ARG)

S1 = SIN(ARG)

C2 = C1**2 - S1**2

S2 = C1*S1 + C1*S1

C3 = C1*C2 - S1*S2

S3 = C2*S1 + S2*C1

C4 = C2**2 - S2**2

S4 = C2*S2 + C2*S2

C5 = C2*C3 - S2*S3

S5 = C3*S2 + S3*C2

C6 = C3**2 - S3**2

S6 = C3*S3 + C3*S3

C7 = C3*C4 - S3*S4

S7 = C4*S3 + S4*C3

INT8 = INT*8

J0 = JR*INT8 + 1

K0 = JI*INT8 + 1

JLAST = J0 + INT - 1

DO 100 J=J0,JLAST

K = K0 + J - J0

TR1 = BR1(J)*C1 - BI1(K)*S1

TI1 = BR1(J)*S1 + BI1(K)*C1

TR2 = BR2(J)*C2 - BI2(K)*S2

TI2 = BR2(J)*S2 + BI2(K)*C2

TR3 = BR3(J)*C3 - BI3(K)*S3

TI3 = BR3(J)*S3 + BI3(K)*C3

TR4 = BR4(J)*C4 - BI4(K)*S4

TI4 = BR4(J)*S4 + BI4(K)*C4

TR5 = BR5(J)*C5 - BI5(K)*S5

TI5 = BR5(J)*S5 + BI5(K)*C5

TR6 = BR6(J)*C6 - BI6(K)*S6

TI6 = BR6(J)*S6 + BI6(K)*C6

TR7 = BR7(J)*C7 - BI7(K)*S7

TI7 = BR7(J)*S7 + BI7(K)*C7

C

T0 = BR0(J) + TR4

T1 = BI0(K) + TI4

C

```
TR4 = BR0(J) - TR4
TI4 = BI0(K) - TI4
T2 = TR1 + TR5
T3 = TI1 + TI5
TR5 = TR1 - TR5
TI5 = TI1 - TI5
T4 = TR2 + TR6
T5 = TI2 + TI6
TR6 = TR2 - TR6
TI6 = TI2 - TI6
T6 = TR3 + TR7
T7 = TI3 + TI7
TR7 = TR3 - TR7
TI7 = TI3 - TI7
```

C

```
TR0 = T0 + T4
TI0 = T1 + T5
TR2 = T0 - T4
TI2 = T1 - T5
TR1 = T2 + T6
TI1 = T3 + T7
TR3 = T2 - T6
TI3 = T3 - T7
T0 = TR4 - TI6
T1 = TI4 + TR6
T4 = TR4 + TI6
T5 = TI4 - TR6
T2 = TR5 - TI7
T3 = TI5 + TR7
T6 = TR5 + TI7
T7 = TI5 - TR7
BR0(J) = TR0 + TR1
BI7(K) = TI0 + TI1
BI6(K) = TR0 - TR1
BR1(J) = TI1 - TI0
BR2(J) = TR2 - TI3
BI5(K) = TI2 + TR3
BI4(K) = TR2 + TI3
BR3(J) = TR3 - TI2
PR = P7*(T2-T3)
PI = P7*(T2+T3)
BR4(J) = T0 + PR
BI3(K) = T1 + PI
BI2(K) = T0 - PR
BR5(J) = PI - T1
PR = -P7*(T6+T7)
PI = P7*(T6-T7)
BR6(J) = T4 + PR
BI1(K) = T5 + PI
BI0(K) = T4 - PR
BR7(J) = PI - T5
100 CONTINUE
JR = JR + 2
JI = JI - 2
IF (JI-JL) 110, 110, 120
110 JI = 2*JR - 1
JL = JR
120 CONTINUE
RETURN
```


C

END

C

C-----
C SUBROUTINE: R4SYN

C RADIX 4 SYNTHESIS
C-----

C

SUBROUTINE R4SYN(INT, B0, B1, B2, B3)

DIMENSION B0(2), B1(2), B2(2), B3(2)

DO 10 K=1,INT

T0 = B0(K) + B1(K)

T1 = B0(K) - B1(K)

T2 = B2(K) + B3(K)

T3 = B2(K) - B3(K)

B0(K) = T0 + T2

B2(K) = T0 - T2

B1(K) = T1 + T3

B3(K) = T1 - T3

10 CONTINUE

RETURN

END

C

C

C-----
C SUBROUTINE: R8SYN

C RADIX 8 SYNTHESIS SUBROUTINE
C-----

C

SUBROUTINE R8SYN(INT, NN, BR0, BR1, BR2, BR3, BR4, BR5, BR6, BR7,

* BI0, BI1, BI2, BI3, BI4, BI5, BI6, BI7)

DIMENSION L(15), BR0(2), BR1(2), BR2(2), BR3(2), BR4(2), BR5(2),

* BR6(2), BR7(2), BI0(2), BI1(2), BI2(2), BI3(2), BI4(2),

* BI5(2), BI6(2), BI7(2)

COMMON /CON1/ P11, P7, P7TWO, C22, S22, P12

EQUIVALENCE (L15,L(1)), (L14,L(2)), (L13,L(3)), (L12,L(4)),

* (L11,L(5)), (L10,L(6)), (L9,L(7)), (L8,L(8)), (L7,L(9)),

* (L6,L(10)), (L5,L(11)), (L4,L(12)), (L3,L(13)), (L2,L(14)),

* (L1,L(15))

L(1) = NN/8

DO 40 K=2,15

IF (L(K-1)-2) 10, 20, 30

10 L(K-1) = 2

20 L(K) = 2

GO TO 40

30 L(K) = L(K-1)/2

40 CONTINUE

P10VN = P11/FLOAT(NN)

J1 = 3

JL = 2

JR = 2

C

DO 120 J1=2,L1,2

DO 120 J2=J1,L2,L1

DO 120 J3=J2,L3,L2

DO 120 J4=J3,L4,L3

DO 120 J5=J4,L5,L4

DO 120 J6=J5,L6,L5

DO 120 J7=J6,L7,L6

DO 120 J8=J7,L8,L7

DO 120 J9=J8,L9,L8

C

```

DO 120 J10=J9,L10,L9
DO 120 J11=J10,L11,L10
DO 120 J12=J11,L12,L11
DO 120 J13=J12,L13,L12
DO 120 J14=J13,L14,L13
DO 120 JTHET=J14,L15,L14
TH2 = JTHET - 2
IF (TH2) 50, 50, 90
50 DO 60 K=1,INT
   T0 = BR0(K) + BR1(K)
   T1 = LR0(K) - BR1(K)
   T2 = BR2(K) + BR2(K)
   T3 = BR3(K) + BR3(K)
   T4 = BR4(K) + BR6(K)
   T6 = BR7(K) - BR5(K)
   T5 = BR4(K) - BR6(K)
   T7 = BR7(K) + BR5(K)
   PR = P7*(T7+T5)
   PI = P7*(T7-T5)
   TT0 = T0 + T2
   TT1 = T1 + T3
   T2 = T0 - T2
   T3 = T1 - T3
   T4 = T4 + T4
   T5 = PR + PR
   T6 = T6 + T6
   T7 = PI + PI
   BR0(K) = TT0 + T4
   BR1(K) = TT1 + T5
   BR2(K) = T2 + T6
   BR3(K) = T3 + T7
   BR4(K) = TT0 - T4
   BR5(K) = TT1 - T5
   BR6(K) = T2 - T6
   BR7(K) = T3 - T7
60 CONTINUE
IF (NN-8) 120, 120, 70
70 K0 = INT*8 + 1
   KL = K0 + INT - 1
   DO 80 K=K0,KL
     T1 = B10(K) + B16(K)
     T2 = B17(K) - B11(K)
     T3 = B10(K) - B16(K)
     T4 = B17(K) + B11(K)
     PR = T3*C22 + T4*S22
     PI = T4*C22 - T3*S22
     T5 = B12(K) + B14(K)
     T6 = B15(K) - B13(K)
     T7 = B12(K) - B14(K)
     T8 = B15(K) + B13(K)
     RR = T8*C22 - T7*S22
     RI = -T8*S22 - T7*C22
     B10(K) = (T1+T5) + (T1+T5)
     B14(K) = (T2+T6) + (T2+T6)
     B11(K) = (PR+RR) + (PR+RR)
     B15(K) = (PI+RI) + (PI+RI)
     T5 = T1 - T5
     T6 = T2 - T6
     B12(K) = P7TWO*(T6+T5)

```

C

```

      BI6(K) = P7TWO*(T6-T5)
      RR = PR - RR
      RI = PI - RI
      BI3(K) = P7TWO*(RI+RR)
      BI7(K) = P7TWO*(RI-RR)
80  CONTINUE
      GO TO 120
90  ARG = TH2*PIOVN
      C1 = COS(ARG)
      S1 = -SIN(ARG)
      C2 = C1**2 - S1**2
      S2 = C1*S1 + C1*S1
      C3 = C1*C2 - S1*S2
      S3 = C2*S1 + S2*C1
      C4 = C2**2 - S2**2
      S4 = C2*S2 + C2*S2
      C5 = C2*C3 - S2*S3
      S5 = C3*S2 + S3*C2
      C6 = C3**2 - S3**2
      S6 = C3*S3 + C3*S3
      C7 = C3*C4 - S3*S4
      S7 = C4*S3 + S4*C3
      INT8 = INT*8
      J0 = JR*INT8 + 1
      K0 = JI*INT8 + 1
      JLAST = J0 + INT - 1
      DO 100 J=J0,JLAST
        K = K0 + J - J0
        TR0 = BR0(J) + BI6(K)
        TI0 = BI7(K) - BR1(J)
        TR1 = BR0(J) - BI6(K)
        TI1 = BI7(K) + BR1(J)
        TR2 = BR2(J) + BI4(K)
        TI2 = BI5(K) - BR3(J)
        TR3 = BI5(K) + BR3(J)
        TI3 = BI4(K) - BR2(J)
        TR4 = BR4(J) + BI2(K)
        TI4 = BI3(K) - BR5(J)
        T0 = BR4(J) - BI2(K)
        T1 = BI3(K) + BR5(J)
        TR5 = P7*(T0+T1)
        TI5 = P7*(T1-T0)
        TR6 = BR6(J) + BI0(K)
        TI6 = BI1(K) - BR7(J)
        T0 = BR6(J) - BI0(K)
        T1 = BI1(K) + BR7(J)
        TR7 = -P7*(T0-T1)
        TI7 = -P7*(T1+T0)
        T0 = TR0 + TR2
        T1 = TI0 + TI2
        T2 = TR1 + TR3
        T3 = TI1 + TI3
        TR2 = TR0 - TR2
        TI2 = TI0 - TI2
        TR3 = TR1 - TR3
        TI3 = TI1 - TI3
        T4 = TR4 + TR6
        T5 = TI4 + TI6
        T6 = TR5 + TR7

```

C

```

T7 = T15 + T17
TTR6 = T14 - T16
T16 = TR6 - TR4
TTR7 = T15 - T17
T17 = TR7 - TR5
BR0(J) = T0 + T4
B10(K) = T1 + T5
BR1(J) = C1*(T2+T6) - S1*(T3+T7)
B11(K) = C1*(T3+T7) + S1*(T2+T6)
BR2(J) = C2*(TR2+TTR6) - S2*(T12+T16)
B12(K) = C2*(T12+T16) + S2*(TR2+TTR6)
BR3(J) = C3*(TR3+TTR7) - S3*(T13+T17)
B13(K) = C3*(T13+T17) + S3*(TR3+TTR7)
BR4(J) = C4*(T0-T4) - S4*(T1-T5)
B14(K) = C4*(T1-T5) + S4*(T0-T4)
BR5(J) = C5*(T2-T6) - S5*(T3-T7)
B15(K) = C5*(T3-T7) + S5*(T2-T6)
BR6(J) = C6*(TR2-TTR6) - S6*(T12-T16)
B16(K) = C6*(T12-T16) + S6*(TR2-TTR6)
BR7(J) = C7*(TR3-TTR7) - S7*(T13-T17)
B17(K) = C7*(T13-T17) + S7*(TR3-TTR7)

```

```

100 CONTINUE
    JR = JR + 2
    JI = JI - 2
    IF (JI-JL) 110, 110, 120
110   JI = 2*JR - 1
    JL = JR
120 CONTINUE
    RETURN
    END

```

C

C-----
C SUBROUTINE: ORD1
C IN-PLACE REORDERING SUBROUTINE
C-----

C

```

SUBROUTINE ORD1(M, B)
  DIMENSION B(2)

```

C

```

    K = 4
    KL = 2
    N = 2**M
    DO 40 J=4,N,2
      IF (K-J) 20, 20, 10
10     T = B(J)
        B(J) = B(K)
        B(K) = T
20     K = K - 2
      IF (K-KL) 30, 30, 40
30     K = 2*J
        KL = J
40 CONTINUE
    RETURN
    END

```

C

C-----
C SUBROUTINE: ORD2
C IN-PLACE REORDERING SUBROUTINE
C-----

C

C

C

```
SUBROUTINE ORD2(M, B)
  DIMENSION L(15), B(2)
  EQUIVALENCE (L15,L(1)), (L14,L(2)), (L13,L(3)), (L12,L(4)),
*   (L11,L(5)), (L10,L(6)), (L9,L(7)), (L8,L(8)), (L7,L(9)),
*   (L6,L(10)), (L5,L(11)), (L4,L(12)), (L3,L(13)), (L2,L(14)),
*   (L1,L(15))
  N = 2**M
  L(1) = N
  DO 10 K=2,M
    L(K) = L(K-1)/2
10  CONTINUE
  DO 20 K=M,14
    L(K+1) = 2
20  CONTINUE
  IJ = 2
  DO 40 J1=2,L1,2
    DO 40 J2=J1,L2,L1
      DO 40 J3=J2,L3,L2
        DO 40 J4=J3,L4,L3
          DO 40 J5=J4,L5,L4
            DO 40 J6=J5,L6,L5
              DO 40 J7=J6,L7,L6
                DO 40 J8=J7,L8,L7
                  DO 40 J9=J8,L9,L8
                    DO 40 J10=J9,L10,L9
                      DO 40 J11=J10,L11,L10
                        DO 40 J12=J11,L12,L11
                          DO 40 J13=J12,L13,L12
                            DO 40 J14=J13,L14,L13
                              DO 40 J1=J14,L15,L14
                                IF (IJ-J1) 30, 40, 40
30   T = B(IJ-1)
      B(IJ-1) = B(J1-1)
      B(J1-1) = T
      T = B(IJ)
      B(IJ) = B(J1)
      B(J1) = T
40   IJ = IJ + 2
  RETURN
  END
```

C

C

C SUBROUTINE: FFT842
C FAST FOURIER TRANSFORM FOR N=2**M
C COMPLEX INPUT

C

C

SUBROUTINE FFT842(IN, N, X, Y)

C

C THIS PROGRAM REPLACES THE VECTOR Z=X+IY BY ITS FINITE
C DISCRETE, COMPLEX FOURIER TRANSFORM IF IN=0. THE INVERSE TRANSFORM
C IS CALCULATED FOR IN=1. IT PERFORMS AS MANY BASE
C 8 ITERATIONS AS POSSIBLE AND THEN FINISHES WITH A BASE 4 ITERATION
C OR A BASE 2 ITERATION IF NEEDED.

C

C THE SUBROUTINE IS CALLED AS SUBROUTINE FFT842 (IN,N,X,Y).
C THE INTEGER N (A POWER OF 2), THE N REAL LOCATION ARRAY X, AND
C THE N REAL LOCATION ARRAY Y MUST BE SUPPLIED TO THE SUBROUTINE.

```

C
C
C      DIMENSION X(2), Y(2), L(15)
C      COMMON /CON2/ P12, P7
C      EQUIVALENCE (L15,L(1)), (L14,L(2)), (L13,L(3)), (L12,L(4)),
*      (L11,L(5)), (L10,L(6)), (L9,L(7)), (L8,L(8)), (L7,L(9)),
*      (L6,L(10)), (L5,L(11)), (L4,L(12)), (L3,L(13)), (L2,L(14)),
*      (L1,L(15))
C
C
C      IW IS A MACHINE DEPENDENT WRITE DEVICE NUMBER
C
C      IW = 11MACH(2)
C      IW=1
C
C      P12 = 8.*ATAN(1.)
C      P7 = 1./SQRT(2.)
C      DO 10 I=1,15
C          M = I
C          NT = 2**I
C          IF (N.EQ.NT) GO TO 20
10  CONTINUE
C      WRITE (IW,9999)
9999 FORMAT (35H N IS NOT A POWER OF TWO FOR FFT842)
C      STOP
C      20  N2POW = M
C          NTHPO = N
C          FN = NTHPO
C          IF (IN.EQ.1) GO TO 40
C          DO 30 I=1,NTHPO
C              Y(I) = -Y(I)
30  CONTINUE
C      40  N8POW = N2POW/3
C          IF (N8POW.EQ.0) GO TO 60
C
C      RADIX 8 PASSES, IF ANY.
C
C      DO 50 IPASS=1,N8POW
C          NXTLT = 2**((N2POW-3*IPASS))
C          LENGT = 8*NXTLT
C          CALL R8TX(NXTLT, NTHPO, LENGT, X(1), X(NXTLT+1), X(2*NXTLT+1),
*          X(3*NXTLT+1), X(4*NXTLT+1), X(5*NXTLT+1), X(6*NXTLT+1),
*          X(7*NXTLT+1), Y(1), Y(NXTLT+1), Y(2*NXTLT+1), Y(3*NXTLT+1),
*          Y(4*NXTLT+1), Y(5*NXTLT+1), Y(6*NXTLT+1), Y(7*NXTLT+1))
50  CONTINUE
C
C      IS THERE A FOUR FACTOR LEFT
C
C      60  IF (N2POW-3*N8POW-1) 90, 70, 80
C
C      GO THROUGH THE BASE 2 ITERATION
C
C      70  CALL R2TX(NTHPO, X(1), X(2), Y(1), Y(2))
C          GO TO 90
C
C      GO THROUGH THE BASE 4 ITERATION
C
C      80  CALL R4TX(NTHPO, X(1), X(2), X(3), X(4), Y(1), Y(2), Y(3), Y(4))
C

```

C

```

90 DO 110 J=1,15
    L(J) = 1
    IF (J-N2POW) 100, 100, 110
100 L(J) = 2** (N2POW+1-J)
110 CONTINUE
    IJ = 1
    DO 130 J1=1,L1
    DO 130 J2=J1,L2,L1
    DO 130 J3=J2,L3,L2
    DO 130 J4=J3,L4,L3
    DO 130 J5=J4,L5,L4
    DO 130 J6=J5,L6,L5
    DO 130 J7=J6,L7,L6
    DO 130 J8=J7,L8,L7
    DO 130 J9=J8,L9,L8
    DO 130 J10=J9,L10,L9
    DO 130 J11=J10,L11,L10
    DO 130 J12=J11,L12,L11
    DO 130 J13=J12,L13,L12
    DO 130 J14=J13,L14,L13
    DO 130 J1=J14,L15,L14
    IF (IJ-J1) 120, 130, 130
120 R = X(IJ)
    X(IJ) = X(J1)
    X(J1) = R
    FI = Y(IJ)
    Y(IJ) = Y(J1)
    Y(J1) = FI
130 IJ = IJ + 1
    IF (IN.EQ.1) GO TO 150
    DO 140 I=1,NTHPO
    Y(I) = -Y(I)
140 CONTINUE
    GO TO 170
150 DO 160 I=1,NTHPO
    X(I) = X(I)/FN
    Y(I) = Y(I)/FN
160 CONTINUE
170 RETURN
    END

```

C

C-----
C SUBROUTINE: R2TX
C RADIX 2 ITERATION SUBROUTINE
C-----

C

```

SUBROUTINE R2TX(NTHPO, CR0, CR1, CI0, CI1)
DIMENSION CR0(2), CR1(2), CI0(2), CI1(2)
DO 10 K=1,NTHPO,2
    R1 = CR0(K) + CR1(K)
    CR1(K) = CR0(K) - CR1(K)
    CR0(K) = R1
    FI1 = CI0(K) + CI1(K)
    CI1(K) = CI0(K) - CI1(K)
    CI0(K) = FI1
10 CONTINUE
RETURN
END

```

C

C

C-----
C SUBROUTINE: R4TX
C RADIX 4 ITERATION SUBROUTINE
C-----

C

```
SUBROUTINE R4TX(NTHP0, CR0, CR1, CR2, CR3, CI0, CI1, CI2, CI3)
  DIMENSION CR0(2), CR1(2), CR2(2), CR3(2), CI0(2), CI1(2), CI2(2),
  * CI3(2)
  DO 10 K=1,NTHP0,4
    R1 = CR0(K) + CR2(K)
    R2 = CR0(K) - CR2(K)
    R3 = CR1(K) + CR3(K)
    R4 = CR1(K) - CR3(K)
    FI1 = CI0(K) + CI2(K)
    FI2 = CI0(K) - CI2(K)
    FI3 = CI1(K) + CI3(K)
    FI4 = CI1(K) - CI3(K)
    CR0(K) = R1 + R3
    CI0(K) = FI1 + FI3
    CR1(K) = R1 - R3
    CI1(K) = FI1 - FI3
    CR2(K) = R2 - FI4
    CI2(K) = FI2 + R4
    CR3(K) = R2 + FI4
    CI3(K) = FI2 - R4
  10 CONTINUE
  RETURN
  END
```

C

C-----
C SUBROUTINE: R8TX
C RADIX 8 ITERATION SUBROUTINE
C-----

C

```
SUBROUTINE R8TX(NXTLT, NTHP0, LENGT, CR0, CR1, CR2, CR3, CR4,
  * CR5, CR6, CR7, CI0, CI1, CI2, CI3, CI4, CI5, CI6, CI7)
  DIMENSION CR0(2), CR1(2), CR2(2), CR3(2), CR4(2), CR5(2), CR6(2),
  * CR7(2), CI1(2), CI2(2), CI3(2), CI4(2), CI5(2), CI6(2),
  * CI7(2), CI0(2)
  COMMON /CON2/ PI2, P7
```

C

```
  SCALE = PI2/LOAT(LENGT)
  DO 30 J=1,NXTLT
    ARG = FLOAT(J-1)*SCALE
    C1 = COS(ARG)
    S1 = SIN(ARG)
    C2 = C1**2 - S1**2
    S2 = C1*S1 + C1*S1
    C3 = C1*C2 - S1*S2
    S3 = C2*S1 + S2*C1
    C4 = C2**2 - S2**2
    S4 = C2*S2 + C2*S2
    C5 = C2*C3 - S2*S3
    S5 = C3*S2 + S3*C2
    C6 = C3**2 - S3**2
    S6 = C3*S3 + C3*S3
    C7 = C3*C4 - S3*S4
    S7 = C4*S3 + S4*C3
  DO 20 K=J,NTHP0,LENGT
```


C

```

AR0 = CR0(K) + CR4(K)
AR1 = CR1(K) + CR5(K)
AR2 = CR2(K) + CR6(K)
AR3 = CR3(K) + CR7(K)
AR4 = CR0(K) - CR4(K)
AR5 = CR1(K) - CR5(K)
AR6 = CR2(K) - CR6(K)
AR7 = CR3(K) - CR7(K)
AI0 = CI0(K) + CI4(K)
AI1 = CI1(K) + CI5(K)
AI2 = CI2(K) + CI6(K)
AI3 = CI3(K) + CI7(K)
AI4 = CI0(K) - CI4(K)
AI5 = CI1(K) - CI5(K)
AI6 = CI2(K) - CI6(K)
AI7 = CI3(K) - CI7(K)
BR0 = AR0 + AR2
BR1 = AR1 + AR3
BR2 = AR0 - AR2
BR3 = AR1 - AR3
BR4 = AR4 - AI6
BR5 = AR5 - AI7
BR6 = AR4 + AI6
BR7 = AR5 + AI7
BI0 = AI0 + AI2
BI1 = AI1 + AI3
BI2 = AI0 - AI2
BI3 = AI1 - AI3
BI4 = AI4 + AR6
BI5 = AI5 + AR7
BI6 = AI4 - AR6
BI7 = AI5 - AR7
CR0(K) = BR0 + BR1
CI0(K) = BI0 + BI1
IF (J.LE.1) GO TO 10
CR1(K) = C4*(BR0-BR1) - S4*(BI0-BI1)
CI1(K) = C4*(BI0-BI1) + S4*(BR0-BR1)
CR2(K) = C2*(BR2-BI3) - S2*(BI2+BR3)
CI2(K) = C2*(BI2+BR3) + S2*(BR2-BI3)
CR3(K) = C6*(BR2+BI3) - S6*(BI2-BR3)
CI3(K) = C6*(BI2-BR3) + S6*(BR2+BI3)
TR = P7*(BR5-BI5)
TI = P7*(BR5+BI5)
CR4(K) = C1*(BR4+TR) - S1*(BI4+TI)
CI4(K) = C1*(BI4+TI) + S1*(BR4+TR)
CR5(K) = C5*(BR4-TR) - S5*(BI4-TI)
CI5(K) = C5*(BI4-TI) + S5*(BR4-TR)
TR = -P7*(BR7+BI7)
TI = P7*(BR7-BI7)
CR6(K) = C3*(BR6+TR) - S3*(BI6+TI)
CI6(K) = C3*(BI6+TI) + S3*(BR6+TR)
CR7(K) = C7*(BR6-TR) - S7*(BI6-TI)
CI7(K) = C7*(BI6-TI) + S7*(BR6-TR)
GO TO 20
CR1(K) = BR0 - BR1
CI1(K) = BI0 - BI1
CR2(K) = BR2 - BI3
CI2(K) = BI2 + BR3
CR3(K) = BR2 + BI3

```

10

C

CI3(K) = BI2 - BR3
TR = P7*(BR5-BI5)
TI = P7*(BR5+BI5)
CR4(K) = BR4 + TR
CI4(K) = BI4 + TI
CR5(K) = BR4 - TR
CI5(K) = BI4 - TI
TR = -P7*(BR7+BI7)
TI = P7*(BR7-BI7)
CR6(K) = BR6 + TR
CI6(K) = BI6 + TI
CR7(K) = BR6 - TR
CI7(K) = BI6 - TI

20 CONTINUE
30 CONTINUE
RETURN
END

7.3 Iterative Solution of Channel Response Program

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

PROGRAM TO CALCULATE THE CHANNEL RESPONSES
FROM SOURCE TO SENSORS BY ITERATION ON THE
FIRST APPROXIMATIONS OF THESE FUNCTIONS
OBTAINED FROM THE ARRAY BY STEERING AND
FOCUSING. EACH ITERATION IMPROVES THE
ACCURACY OF THE FUNCTIONS BY REDUCING THE
AMOUNT OF ERROR IN EACH APPROXIMATION.

COMPILE: FTM HCONV -64V
LOAD: SEC

LO #HCONV
LI B_HCONV
LI B_MR1DF
LI
SA
Q

RUN: 0 D.1 1 2 (IDEAL H11(W))
0 D.2 2 2 (IDEAL H21(W))
0 D.3 3 2 (INITIAL APPROX H11(W))
0 D.4 4 2 (INITIAL APPROX H21(W))
0 D.5 7 2 (H11(W) AFTER 5 ITERATIONS)
0 D.6 10 2 (H21(W) AFTER 5 ITERATIONS)
0 D.7 11 2 (H11(W) AFTER 10 ITERATIONS)
0 D.8 12 2 (H21(W) AFTER 10 ITERATIONS)
0 D.9 13 2 (H11(W) AFTER 20 ITERATIONS)
0 D.10 14 2 (H21(W) AFTER 20 ITERATIONS)

SEC #HCONV
C 1 2 3 4 7
C 10 11 12 13 14

PLOT: SEC #PLOT2D
D.1
SEC #PLOT2D
D.2
.
.
.

REAL*8 H11(2048),I11(2048),RR(2048),TH(2048),RM
REAL*8 HH11(2048),II11(2048),HH21(2048),II21(2048)
REAL*8 H21(2048),I21(2048)
REAL*8 DEXP,DBLE,DLOG10
COMMON PI,TWOPI,THLINC,THLCON,NFFT,NPTS,NN,L,H,H1,DVTM2
COMMON/A1/H11,I11,H21,I21
COMMON/A2/HH11,II11,HH21,II21
COMMON/A3/RR,TH
LOGICAL ISSUC
READ(1,300)P,Q,R,S
LOG2N=11
N=2**LOG2N
PI=4.*ATAN(1.)
TWOPI=2.*PI
THLINC=1.5
THLCON=.5
NFFT=N
SIGN=-1.
ND2=N/2
M=ND2+1

C CALCULATE TIME DELAYS FOR H11(W) & H21(W)

C

```
T1=SQRT(.125**2+.3**2)
T2=SQRT(.125**2+.5**2)
T3=SQRT(.575**2+.35**2)
T4=SQRT(.575**2+.45**2)
```

C DEFINE IDEAL IMPULSE RESPONSE OF CHANNELS

```
DO 1 I=1,N
  H11(I)=0.D0
  I11(I)=0.D0
  H21(I)=0.D0
1 I21(I)=0.D0
  I=IFIX(T1*1024.+.5)
  H11(I)=1./T1
  I=IFIX(T2*1024.+.5)
  H11(I)=1./T2
  I=IFIX(T3*1024.+.5)
  H21(I)=1./T3
  I=IFIX(T4*1024.+.5)
  H21(I)=1./T4
```

C TAKE FFT OF TIME SIGNAL

```
CALL MR1DF(LOG2N,H11,I11,SIGN)
```

C CONVERT TO POLAR FORM

```
CALL RP(H11,I11,RR,TH,N,0)
```

C OUTPUT FREQUENCY DOMAIN (MAGNITUDE)

```
CALL WRI(RR,RR,1024,5)
```

```
300 FORMAT(G20.14)
```

```
CALL MR1DF(LOG2N,H21,I21,SIGN)
```

C CONVERT TO POLAR FORM

```
CALL RP(H21,I21,RR,TH,N,0)
```

```
CALL WRI(RR,RR,1024,6)
```

C

C CALCULATE ACTUAL RESPONSE RECEIVED BY ARRAY

C TO BE USED AS INITIAL APPROXIMATIONS

```
DO 2 I=1,N
  HH11(I)=P*H11(I)+Q*H21(I)
  II11(I)=P*I11(I)+Q*I21(I)
  HH21(I)=R*H11(I)+S*H21(I)
  II21(I)=R*I11(I)+S*I21(I)
2
```

C CONVERT TO POLAR FORM AND OUTPUT

```
CALL RP(HH11,II11,RR,TH,N,0)
```

```
CALL WRI(RR,RR,1024,7)
```

```
CALL RP(HH21,II21,RR,TH,N,0)
```

```
CALL WRI(RR,RR,1024,8)
```

C

C SET UP FOR ITERATIVE PROCEDURE

```
DO 3 I=1,N
  H11(I)=HH11(I)/P
  I11(I)=II11(I)/P
3
K=10
```

C ITERATE ON RESPONSE FUNCTIONS USING THE

C IMPROVED ACCURACY OF EACH FUNCTION TO REDUCE

C THE ERROR IN ITS COMPLIMENT (H11(W) AND H21(W))

```
DO 4 J=1,20
DO 5 I=1,N
  H21(I)=HH21(I)/S-R*H11(I)/S
  I21(I)=II21(I)/S-R*I11(I)/S
  H11(I)=HH11(I)/P-Q*H21(I)/P
  I11(I)=II11(I)/P-Q*I21(I)/P
5
IF(.NOT.(J.EQ.5.OR.J.EQ.10.OR.J.EQ.20))GO TO 4
K=K+1
```

```
C  CONVERT TO POLAR FORM AND OUTPUT
C  SELECTED ITERATIONS
    CALL RP(H11,I11,RR,TH,N,0)
    CALL WRI(RR,RR,1024,K)
    K=K+1
    CALL RP(H21,I21,RR,TH,N,0)
    CALL WRI(RR,RR,1024,K)
4  CONTINUE
    CALL EXIT
    END
```

7.4 Dual Source Imaging Program

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

PROGRAM TO IMAGE SOURCE PLANE CONTAINING ACOUSTIC
SOURCES BY CORRELATION TECHNIQUES ON THE SIGNALS
RECEIVED AT THE ARRAY ELEMENTS MONITORING THE PLANE.
THIS PROGRAM SIMULATES THE RECEIVED SIGNALS AND USES
CROSS-CORRELATION PROCESSING TO CREATE A 2-D SURFACE
PLOT DATA FOR THE LOCATION AND POWER OF THE
SOURCES IN THE PLANE.
2 SOURCES EACH WITH AN ECHO ARE SIMULATED
WITH VARIOUS CHANGES MADE TO DEMONSTRATE THE
EFFECTS OF DIFFERENT PROCESSING TECHNIQUES
INCLUDING DEREVERBERATION.

COMPILE: FTM ACORR -64V

LOAD: SEC

LO #ACORR

LO B_ACORR

LI

SA

Q

RUN: 0 D.1 1 2

(SIGNALS RECEIVED AT EACH SENSOR)

0 D.2 2 2

(2-D SOURCE POWER DATA)

SEC #ACORR

C 1 2

PLOT: SEC #PLOT2D

D.1

SEC #PLOT3D

D.2

0

0

3

30.

.8

50.

10.

30.

0

0

REAL*4 X(4,5120),XX(64,64),CX(4,4),CY(4),T(4),PX(4),IS(4)

REAL*4 TT(4,4),TAU(4)

N=2048

II=4

XMIN=0.

XMAX=1.

C DEFINE X,Y LOCATIONS OF SOURCES AND ECHOS

PX(1)=.125

PX(2)=.375

PX(3)=.625

PX(4)=.875

CX(1,1)=.125*.125

CX(1,3)=.575*.575

CX(2,1)=.125*.125

CX(2,3)=.325*.325

CX(3,1)=.375*.375

CX(3,3)=.075*.075

CX(4,1)=.625*.625

CX(4,3)=.175*.175

CY(1)=.3*.3

C

```
      TMIN=2.  
      TMAX=0.  
      DO 8 K=1,4  
        T(K)=SQRT((X1-PX(K))**2+XJ*XJ)  
        IF(T(K).GE.TMIN)GO TO 8  
        TMIN=T(K)  
CC 7    IF(T(K).LE.TMAX)GO TO 8  
CC      TMAX=T(K)  
      8    CONTINUE  
      DO 9 K=1,4  
C    CALCULATE TIME SHIFT VALUE  
      9    IS(K)=IFIX((T(K)-TMIN)*2048.+.5)  
      S=0.  
C    INTEGRATE SHIFTED PRODUCT SIGNAL  
      DO 10 K=1,3072  
        SS=0.  
        DO 11 L=1,4  
          KK=K+IS(L)  
11      SG=SG+X(L,KK)  
C    ADD UP POWERS FROM ALL SIGNALS  
10      S=S+SG*SG  
C    SCALE INTEGRAL VALUE  
      XX(1,J)=S/2048.  
C    OUTPUT POWER VALUE A THAT POINT IN THE PLANE  
      WRITE(6,800)XX(1,J)  
800    FORMAT(E15.8)  
      5    CONTINUE  
400    FORMAT(2I3,4G15.8)  
      STOP  
      END
```

7.5 Fresnel Diffraction Array Simulation Program

C *** ULTRASOUND 1 & 2 DIMENSIONAL ARRAY PATTERN PROGRAM ***

C *** ULTRASOUND 1 & 2 DIMENSIONAL ARRAY PATTERN PROGRAM ***

C COMPILE: FTSN FRSNL
C LOAD:
C LOAD
C LO B_FRSNL
C LI
C SA *FRSNL
C QU

C \$INSERT SYSCOM>KEYS.F

C Z = DISTANCE FROM ELEMENT PLANE (METERS) EX .018
C DZ = INCREMENT TO Z FOR EACH PLOT OF MM EX .0005
C NXEL = NUMBER OF ELEMENTS IN X DIRECTION EX 10
C NYEL = NUMBER OF ELEMENTS IN Y DIRECTION EX 10
C NZ = NUMBER OF PLOTS GENERATED FOR DIFFERENT Z'S EX 20
C NA = NUMBER OF PLOTS GENERATED FOR DIFFERENT A'S EX 20
C KXPTS = NUMBER OF POINTS IN X DIRECTION ON PLOTTING SURFACE EX 50
C KYPTS = NUMBER OF POINTS IN Y DIRECTION ON PLOTTING SURFACE EX 50
C XD = CENTER TO CENTER SPACING OF ELEMENTS: X DIRECTION (METERS) EX .001
C YD = CENTER TO CENTER SPACING OF ELEMENTS: Y DIRECTION (METERS) EX .001
C L = HALF THE WIDTH OF ELEMENT (METERS) EX .000125
C VS = SPEED OF SOUND IN WATER (METERS/SEC) EX 1400.
C AMP(II,JJ) = GAIN ON ELEMENT(II,JJ) EX 1.20
C XDEC = MAXIMUM VALUE OF LINEAR PHASE SHIFT, X DIRECT EX 5.
C YDEC = MAXIMUM VALUE OF LINEAR PHASE SHIFT, Y DIRECT EX 5.
C FOCUS = MAXIMUM VALUE OF QUADRATIC PHASE SHIFT EX 5.
C XTIAX = MAXIMUM X-DIMENSION OF VIEWING AREA (METERS) EX +.015
C XMIN = MINIMUM X-DIMENSION OF VIEWING AREA (METERS) EX -.015
C YMAX = MAXIMUM Y-DIMENSION OF VIEWING AREA (METERS) EX +.015
C YMIN = MINIMUM Y-DIMENSION OF VIEWING AREA (METERS) EX -.015

COMMON /AREA1/ J,XC,YC,NXEL,NYEL,PI,V0,V6,APHI,PI2
REAL AMP(20),PHI0,L,LAMBDA,K,XC(50,20),YC(50,20),Z,ZI
REAL ARRAY(410,20)
COMPLEX J,VAL,U,CEXP,CSQRT,V0,V2,V6,APHI(50,20),CEXP0
DOUBLE PRECISION PI2

700 FORMAT(8G15.8)
600 FORMAT(213,2G15.8)
300 FORMAT(212,2G15.8)
PI=3.1415926535
PI2=6.28318530717958647
J=(0.,1.)
READ(7,700)NF
READ(7,700)F
READ(7,700)DF
READ(7,700)VS
READ(7,700)KXPTS
READ(7,700)KYPTS
READ(7,700)XTIAX
READ(7,700)XMIN
READ(7,700)YMAX
READ(7,700)YMIN
READ(7,700)XD
READ(7,700)YD
READ(7,700)L
READ(7,700)NZ
READ(7,700)Z
READ(7,700)DZ
READ(7,700)FOCUS
READ(7,700)XDEC
READ(7,700)YDEC
READ(7,700)SWT

C *** ULTRASOUND 1 8 2 DIMENSIONAL ARRAY PATTERN PROGRAM ***

```

XSTEP=0.
IF(KXPTS.EQ.1)GO TO 4
XSTEP=(XMAX-XMIN)/FLOAT(KXPTS-1)
4 YSTEP=0.
IF(KYPTS.EQ.1)GO TO 3
YSTEP=(YMAX-YMIN)/FLOAT(KYPTS-1)
3 XI2=1./L**2
READ(6,700)XN
NA=IFIX(XN)
F=F-DF
DO 2 M=1,NF
F=F+DF
LAMBDA=VS/F
K=2.*PI/LAMBDA
DO 2 N=1,NA
READ(6,700)XN
NXEL=IFIX(XN)
READ(6,700)XN
NYEL=IFIX(XN)
DO 7 II=1,NXEL
READ(6,700)(AMP(JJ),JJ=1,NYEL)
DO 7 JJ=1,NYEL
APHI(II,JJ)=CMPLX(AMP(JJ),0.)
7 CONTINUE
IF(FOCUS.EQ.0.)GO TO 30
FOCUS=-K/2./FOCUS
30 XDEC=K*SIN(PI*XDEC/180.)/COS(PI*XDEC/180.)
YDEC=K*SIN(PI*YDEC/180.)/COS(PI*YDEC/180.)
IF(XD.EQ.0.)GO TO 5
X0=XD*(NXEL-1)/2.
DO 50 II=1,NXEL
DO 50 JJ=1,NYEL
XC(II,JJ)=(II-1)*XD-X0
50 CONTINUE
GO TO 8
5 CONTINUE
READ(8,700)XN
READ(8,700)XN
READ(8,700)XN
DO 90 II=1,NXEL
90 READ(8,700)(XC(II,JJ),JJ=1,NYEL)
8 IF(YD.EQ.0.)GO TO 6
Y0=YD*(NYEL-1)/2.
DO 60 II=1,NXEL
DO 60 JJ=1,NYEL
YC(II,JJ)=(JJ-1)*YD-Y0
60 CONTINUE
GO TO 9
6 CONTINUE
DO 80 II=1,NXEL
80 READ(8,700)(YC(II,JJ),JJ=1,NYEL)
9 IF(SWT.EQ.0.)GO TO 1
DO 1 II=1,NXEL
DO 1 JJ=1,NYEL
R=XC(II,JJ)
THETA=YC(II,JJ)
XC(II,JJ)=R*COS(THETA)
YC(II,JJ)=R*SIN(THETA)
1 CONTINUE

```

C *** ULTRASOUND 1 & 2 DIMENSIONAL ARRAY PATTERN PROGRAM ***

```

      DO 70 II=1,NXEL
      DO 70 JJ=1,NYEL
C   SET FOR LINEAR AND QUADRATIC PHASING
      PHI0=FOCUS*(XC(II,JJ)*XC(II,JJ)+YC(II,JJ)*YC(II,JJ))
      8      +XDEG*(XC(II,JJ)-XC(1,1))
      8      +YDEG*(YC(II,JJ)-YC(1,1))
      APHI(II,JJ)=APHI(II,JJ)*CEXP0(J*PHI0)
C   PHI0=PHI0+PHI(II,JJ)
70    CONTINUE
      Z=Z-DZ
      DO 2 I=1,NZ
      Z=Z+DZ
      V0=I*K*Z
      V2=-J*K/Z
      V0=V2*CEXP0(V0)/(XI2+V2)
      V6=-V2/2./(1.+V2/XI2)
      DO 10 JJ=1,KYPTS
      Y0=YSTEP*(JJ-1)+YMIN
      DO 11 II=1,KXPTS
      X0=XSTEP*(II-1)+XMIN
      VAL=U(X0,Y0)
      ARRAY(II,(1+(N-1)*NZ))=(CABS(VAL))**2
11    CONTINUE
10    CONTINUE
2     CONTINUE
      N=NZ*NA*NF
      WRITE(5,600)KXPTS,N,XMIN,XMAX
      WRITE(5,1000)
      FORMAT(' (8E12.5)')
1000  WRITE(5,900)((ARRAY(II,I),I=1,N),II=1,KXPTS)
900   FORMAT(8E12.5)
      CALL EXIT
      END

```

C
C

```

      FUNCTION U(X0,Y0)
      COMMON /AREA1/ J,XC,YC,NXEL,NYEL,PI,V0,V6,APHI,PI2
      REAL XC(50,20),YC(50,20),X0,Y0
      DOUBLE PRECISION PI2
      COMPLEX U,J,V0,V4,V6,SUM,CEXP,APHI(50,20),CEXP0
      SUM=(0.,0.)
      DO 1 II=1,NXEL
      DO 1 JJ=1,NYEL
      IF(REAL(APHI(II,JJ)).EQ.0.)GO TO 1
      V4=V6*((X0-XC(II,JJ))**2+(Y0-YC(II,JJ))**2)
C   IF(REAL(V4).LT.-87.)GO TO 1
      SUM=SUM+APHI(II,JJ)*CEXP0(V4)
1     CONTINUE
      U=V0*SUM
      RETURN
      END

```

C
C

```

      COMPLEX FUNCTION CEXP0(Z)
      COMMON /AREA1/ J,XC,YC,NXEL,NYEL,PI,V0,V6,APHI,PI2
      REAL XC(50,20),YC(50,20),PI
      COMPLEX CEXP,CEXP0,J,Z,ZZ,V0,V6,APHI(50,20)
      DOUBLE PRECISION Y,PI2
      CEXP0=CEXP(Z)

```

C *** ULTRASOUND 1 & 2 DIMENSIONAL ARRAY PATTERN PROGRAM ***

```
      RETURN
99    Y=DBLE(AIMAG(Z))
      IF(DABS(Y).LT.51000.)GO TO 1
      X=REAL(Z)
      Y=Y-DINT(Y/PI2)*PI2
      ZZ=X+J*SNGL(Y)
      CEXP0=CEXP(ZZ)
      RETURN
1     CEXP0=CEXP(Z)
      RETURN
      END
```

7.6 Minimax Array Optimization Program

```

C
C
C
C
C      MAIN FOR MINI1W OPTIMIZING FRESNEL DIFFRACTION
C      IW=2*N*M+N*N+14*N+4*M+11-5*N-M-5=2NM+NN+9N+3M+6
      IMPLICIT REAL*4 (A-H,O-Z), INTEGER*2 (I-N)
      REAL*8 X,W,DX,EPS
      INTEGER*4 IREF,N,M,MAXFUN,IW,NUIREF
      DIMENSION X(38),W(26740),IREF(487)
      DIMENSION YC(32,32),PHI(32,32)
      COMPLEX APHI(32,32),V0,V6
      COMMON /AREA0/ KXPTS,KYPTS,KZPTS,NXEL,NYEL
      COMMON /AREA1/ YC,PHI,V0,V6,APHI
      COMMON /AREA3/ W,IREF
      EXTERNAL FDF
C      GAINS: X(1)->X(29), SPACINGS: X(30)->X(58)
      KXPTS=191
      KYPTS=1
      KZPTS=1
      READ(8,30)X1,X2,X3
      NXEL=ifix(X2)
      NYEL=ifix(X3)
      DO 1 I=1,NXEL
1      READ(8,300)(X((I-1)*NYEL+J),J=1,NYEL)
300  FORMAT(8G15.8)
      NX1=NXEL*NYEL+1
      N=2*NXEL*NYEL
      READ(9,30)X1,X2,X3
      DO 2 I=1,NXEL
2      READ(9,300)(X((I-1)*NYEL+J+NX1),J=1,NYEL)
      N=4*NXEL*NYEL IF OPTIMIZING YC & PHI, OTHERWISE N=2*NXEL*NYEL
C      M=KXPTS*KYPTS*KZPTS
C      M=KXPTS*KYPTS*KZPTS
      DX=.012D00
      EPS=1.D-14
      MAXFUN=100
      IW=2*N*M+N*N+9*N+3*M+6
      NUIREF=5*N+M+5
C
C
C      FILLING UP ARRAY
      DO 5 I=1,NXEL
      DO 5 J=1,NYEL
        YC(I,J)=0.E00
        PHI(I,J)=0.E00
5      CONTINUE
      CALL MINI1W(FDF,N,M,X,DX,EPS,MAXFUN,W,IW,IREF,NUIREF)
C      * FDF FUNCTION CALLS = W(M+1)
      WRITE(1,10) (X(I),I=1,N)
10  FORMAT(2(' SOLUTION ',4F20.14/6(10X,4F20.14/),10X,F20.14/,/)/)
      WRITE(1,40)W(M+1)
40  FORMAT(/' CALLS OF FDF ',G15.8/)
      XMAX=1.5
      XMIN=-XMAX
      WRITE(5,100)M,NYEL,XMIN,XMAX
100  FORMAT(2I3,2G15.8)
      WRITE(5,200)
200  FORMAT(' (E15.8)')
      WRITE(5,20)(W(I),I=1,M)
20  FORMAT(E15.8)

```


C

```

WRITE(6,30)NYEL,NXEL,NYEL,(X(I),I=1,NXEL),W(M+1)
WRITE(7,30)NYEL,NXEL,NYEL
WRITE(7,30)(X(I),I=NX1,N),(YC(I,1),I=1,NXEL)
30 FORMAT(G15.8)
CALL EXIT
END

```

C
C
C

SUBROUTINE FDF(N,M,X,A,B)

C
C
C

FRESNEL DIFFRACTION PROGRAM & DERIVATIVES

```

IMPLICIT REAL*4 (A-H,O-Z), INTEGER*2 (I-N)
REAL*8 X,A,B,BMAX
INTEGER*4 N,M
COMPLEX J0,VAL,U,CEXP,V0,V2,V6,APHI(32,32)
REAL*4 L,LAMBDA,K,KK(192,1),K1,K2,KL2,K2L2
INTEGER*2 P,Q
DIMENSION X(N),A(M,N),B(M),YC(32,32),PHI(32,32)
DIMENSION R(192,1,58,1),RR(192,1,58,1),CTHET(192,1,58,1),
& STHET(192,1,58,1)
DIMENSION SRCOS(192,1),SRSIN(192,1),SRCOS2(192,1),SRSIN2(192,1)
COMMON /AREA0/ IXPTS,KYPTS,KZPTS,NXEL,NYEL
COMMON /AREA1/ YC,PHI,V0,V6,APHI
COMMON /AREA2/ R,RR,CTHET,STHET,SRCOS,SRSIN,SRCOS2,SRSIN2

```

C
C
C

FRENSL PROGRAM

```

PI=3.1415926535897632
J0=(0.,1.)
F=15000.E00
VS=1524.E00
LAMBDA=VS/F
K=2.*PI/LAMBDA
XMAX=1.5E00
XMIN=-XMAX
YMAX=0.E00
YMIN=-YMAX
L=.005E00
Z1=3.0E00
DZ=.001E00
ANMAX=3.0E00
AXMAX=0.E00
AYMAX=0.E00
XSTEP=0.E00
IF(KKPTS.EQ.1) GO TO 4
XSTEP=(XMAX-XMIN)/(KKPTS-1)
4 YSTEP=0.E00
IF(KYPTS.EQ.1) GO TO 3
C YSTEP=(YMAX-YMIN)/(KYPTS-1)
C
3 XI2=1./L**2
IF(ANMAX.EQ.0.)GO TO 30
ANMAX=-K/2./ANMAX
30 CONTINUE
CC AXMAX=K*DSIN(PI*AXMAX/180.)/DCOS(PI*AXMAX/180.)
CC AYMAX=K*DSIN(PI*AYMAX/180.)/DCOS(PI*AYMAX/180.)
DO 70 I=1,NXEL
DO 70 J=1,NYEL

```

```

C
C   SET FOR LINEAR AND QUADRATIC PHASING
CC   PHIO=AMHAX*(X(I+(J-1)*NXEL+NXEL*NYEL)**2+YC(I,J)**2)
CC   8   +AKHAX*(X(I+(J-1)*NXEL+NXEL*NYEL)-X(I+NXEL*NYEL))
CC   8   +AYHAX*(YC(I,J)-YC(I,1))
C
C   APHI(I,J)=CMPLX(SNCL(X(I)),0.)
C
70  CONTINUE
    Z=Z1-DZ
    DO 2 IZ=1,KZPTS
      Z=Z+DZ
      V0=J0*K*Z
      V2=-J0*K/Z
      V0=V2*CEXP(V0)/(X12+V2)
      V6=-V2/2./(1.+V2*L*L)
      DO 10 J=1,KYPTS
        Y0=YSTEP*(J-1)+YMIN
        DO 11 I=1,KXPTS
          X0=XSTEP*(I-1)+XMIN
          VAL=U(X0,Y0,X,N)
          B(I+(J-1)*KXPTS)=(CABS(VAL))
11     CONTINUE
10    CONTINUE
2     CONTINUE
C
C   FRESNEL DERIVATIVES PROGRAM
C
C   ARRAY(KXPTS,KYPTS)=>B(M) ; M=KXPTS*KYPTS
C   ARRAY(I,J)=>B(I+(J-1)*KXPTS)
C   AMP(NXEL,NYEL)&XC(NXEL,NYEL)&YC(NXEL,NYEL)&PHI(NXEL,NYEL)=>X(N)
C   ; N=4*NXEL*NYEL
C   AMP(I,J)=>X(I+(J-1)*NXEL)
C   XC(I,J)=>X(I+(J-1)*NXEL+NXEL*NYEL)
C   YC(I,J)=>X(I+(J-1)*NXEL+2*NXEL*NYEL)
C   PHI(I,J)=>X(I+(J-1)*NXEL+3*NXEL*NYEL)
C   DAMP(KXPTS,KYPTS,NXEL,NYEL)&DXC()&DYC()&DPHI()=>A(M,N)
C   ; M=KXPTS*KYPTS , N=4*NXEL*NYEL
C   DAMP(I,J,K,L)=>A(I+(J-1)*KXPTS,K+(L-1)*NXEL)
C   DXC(I,J,K,L)=>A(I+(J-1)*KXPTS,K+(L-1)*NXEL+NXEL*NYEL)
C   DYC(I,J,K,L)=>A(I+(J-1)*KXPTS,K+(L-1)*NXEL+2*NXEL*NYEL)
C   DPHI(I,J,K,L)=>A(I+(J-1)*KXPTS,K+(L-1)*NXEL+3*NXEL*NYEL)
C
    KL2=K*L*L
    ZK=Z+KL2*KL2
    K2L2=-K*KL2/2./ZK
    K1=K*Z/2./ZK
    K2=K/ZK
    DO 50 I=1,KXPTS
      X0=XSTEP*(I-1)+XMIN
      DO 50 J=1,KYPTS
        Y0=YSTEP*(J-1)+YMIN
        J=1
        Y0=0.E00
      Z=Z1-DZ
      DO 50 IZ=1,KZPTS
        Z=Z+DZ
        ZK=Z+KL2*KL2
        K2L2=-K*KL2/2./ZK

```

C

```

CC      K1=K*Z/2./ZK
        SRCOS(I,J)=0.
        SRSIN(I,J)=0.
        DO 50 P=1,NXEL
        DO 60 Q=1,NYEL
            XYM=(X0-X(P+(Q-1)*NXEL+NXEL*NYEL))*2+(Y0-YC(P,Q))*2
            THET=PHI(P,Q)+K1*XYM
            CTHET(I,J,P,Q)=COS(THET)
            STHET(I,J,P,Q)=SIN(THET)
            R(I,J,P,Q)=EXP(K2L2*XYM)
            RR(I,J,P,Q)=X(P+(Q-1)*NXEL)*R(I,J,P,Q)
            SRCOS(I,J)=SRCOS(I,J)+RR(I,J,P,Q)*CTHET(I,J,P,Q)
            SRSIN(I,J)=SRSIN(I,J)+RR(I,J,P,Q)*STHET(I,J,P,Q)
60      CONTINUE
        SRCOS2(I,J)=SRCOS(I,J)*SRCOS(I,J)
        SRSIN2(I,J)=SRSIN(I,J)*SRSIN(I,J)
        KK(I,J)=KL2/SQRT(7K)/SQRT(SRCOS2(I,J)+SRSIN2(I,J))
50      CONTINUE

```

```

C      CALCULATE DAMP & DXC
        DO 55 I=1,KXPTS
            X0=XSTEP*(I-1)+XMIN
CC      DO 55 J=1,KYPTS
            Y0=YSTEP*(J-1)+YMIN
CC      Z=Z1-DZ
CC      DO 55 IZ=1,KZPTS
            Z=Z+DZ
CC      ZK=Z*Z+KL2*KL2
CC      K2=K/ZK
        DO 55 P=1,NXEL
        DO 55 Q=1,NYEL
            A(I+(J-1)*KXPTS,P+(Q-1)*NXEL)=KK(I,J)*(SRCOS2(I,J)*R(I,J,P,Q)
8            *CTHET(I,J,P,Q)+SRSIN2(I,J)*R(I,J,P,Q)*STHET(I,J,P,Q))
            A(I+(J-1)*KXPTS,P+(Q-1)*NXEL+NXEL*NYEL)=K2*KK(I,J)*(X0-
8            X(P+(Q-1)*NXEL+NXEL*NYEL))*(SRCOS(I,J)*RR(I,J,P,Q)*
8            (Z*STHET(I,J,P,Q)+KL2*CTHET(I,J,P,Q))+SRSIN(I,J)*
8            RR(I,J,P,Q)*(-Z*CTHET(I,J,P,Q)+KL2*STHET(I,J,P,Q)))
55      CONTINUE

```

```

C      CALCULATE      DYC
C      CALCULATE      DPHI
C

```

```

C      NORMALIZE RESPONSE
        BMAX=B(1)
        DO 90 I=1,KXPTS
90      BMAX=DMAX1(BMAX,B(I))
        MID=KXPTS/2+1
        IWID=3
        IS=MID-IWID
        IF=MID+IWID
        DO 92 I=IS,IF
92      B(I)=B(I)-BMAX*DBLE(COS(FLOAT(I-MID)/FLOAT(IWID)*1.57))
        RETURN
        END

```

C
C

```

FUNCTION U(X0,Y0,X,N)
IMPLICIT REAL*4 (A-H,O-Z), INTEGER*2 (I-N)
REAL*8 X
INTEGER*4 N
COMPLEX U,V0,V4,V6,SUM,CEXP,APHI(32,32)

```

C

```

DIMENSION YC(32,32),PHI(32,32),X(N)
COMMON /AREA0/ KXPTS,KYPTS,KZPTS,NXEL,NYEL
COMMON /AREA1/ YC,PHI,V0,V6,APHI
SUM=(0.,0.)
DO 1 I=1,NXEL
DO 1 J=1,NYEL
  V4=V6*((X0-SNCL(X(I+(J-1)*NXEL+NXEL*NYEL))**2+(Y0-YC(I,J))**2)
  IF(REAL(V4).LT.-1.E05)GO TO 1
  SUM=SUM+APHI(I,J)*CEXP(V4)
1 CONTINUE
U=V0*SUM
RETURN
END

```

C
C
C
C

SUBROUTINE MINI1W(FDF,N,M,X,DX,EPS,MAXFUN,W,IW,IREF,NUIREF)

C
C
C
C
C
C
C

NON-LINEAR MINIMAX OPTIMIZATION USING GRADIENTS.

FOR A PROGRAM DESCRIPTION SEE:

K. MADSEN AND H. SCHJAER-JACOBSEN, 'FORTRAN SUBROUTINES FOR
NON-LINEAR MINIMAX OPTIMIZATION', REPORT R135, ELECTROMAGNETICS
INSTITUTE, TECHNICAL UNIVERSITY OF DENMARK, FEBRUARY 1975.

```

IMPLICIT REAL*8 (A-H,O-Z), INTEGER*4 (I-N)
DIMENSION X(N),W(IW),IREF(NUIREF)
EXTERNAL FDF
IF ((N.LT.1).OR.(M.LT.1).OR.(DX.LE.0D0).OR.(EPS.LE.0D0))RETURN
N1=N+1
NB=1
NB1=NB+M
NA=NB1+M
NA1=NA+N*M
NXX=NA1+M*N
NY=NXX+N
NUY=N1*(N1+5)+M
NIREF=NY+NUY
NUIREF=5*N1+M
CALL SUB1W(FDF,N,M,X,DX,EPS,W(NA),W(NA1),W(NB),W(NB1),W(NXX),
IW(NY),IREF,MAXFUN,NUY,NUIREF)
RETURN
END

```

SUBROUTINE SUB1W(FDF,N,M,X,DX,EPS,A,A1,B,B1,XX,Y,IREF,MAXIT,
NUY,NUIREF)

```

IMPLICIT REAL*8 (A-H,O-Z), INTEGER*4 (I-N)
DIMENSION X(N),A(M,N),A1(M,N),B(M),B1(M),XX(N),Y(NUY)
INTEGER*4 IREF(NUIREF)
LOGICAL DIV4

```

C

THE FOLLOWING NUMBER IS AN EXPRESSION FOR THE MACHINE ACCURACY

C

SEPS=16.**(-13)
SET SOME CONSTANTS

DIV4=.FALSE.

NTAL=0

M1=N+1

C

FIND THE LENGTH OF THE STARTING VECTOR

XMAX=0D0

DO 10 I=1,N

```

C
10 XMAX=DMAX1(XMAX, DABS(X(1)))
C   CALCULATE FUNCTION VALUES
   CALL FDF(N,M,X,A,B)
   F0=0D0
   DO 20 J=1,M
   S= DABS(B(J))
20 IF (S.GT.F0) F0=S
C   ITERATIVE LOOP STARTS HERE
C   FIND THE SOLUTION XX TO THE LINEAR SUBPROBLEM
30 CALL MA19W (N,M,A,M,B,DX,0D0,XX,Y,IREF,NUY,NUIREF)
   F=Y(M1)
   IF (F.GT.F0) GOTO 139
C   FIND THE NEW POINT
   DX=0D0
   DO 40 I=1,N
   S= DABS(XX(I))
   IF (S.GT.DX) DX=S
40 Y(I)=X(I)+XX(I)
C   CALCULATE FUNCTION VALUES
   CALL FDF(N,M,Y,A1,B1)
   F1=0D0
   DO 50 J=1,M
   S= DABS(B1(J))
50 IF (S.GT.F1) F1=S
C   REVISE THE STEP LENGTH
   IF ((F0-F1).GT.(F0-F)/4.) GOTO 60
   DX=DX/4.
   DIV4=.TRUE.
   GOTO 100
60 IF (DIV4) GOTO 90
   S=0D0
   DO 80 J=1,M
   T=B(J)
   DO 70 I=1,N
   T=T+A(J,I)*XX(I)
   T= DABS(B1(J)-T)
70 IF (T.GT.S) S=T
80 IF (S.LE.(F0-F1)/4.) DX=DX*2.
90 DIV4=.FALSE.
C   TEST IF THE NEW POINT IS ACCEPTABLE
100 IF ((F0-F1).LE.(.01*(F0-F))) GOTO 130
C   INTRODUCE THE NEW POINT
   F0=F1
   XMAX=0D0
   DO 110 I=1,N
   X(I)=Y(I)
   XMAX=DMAX1(XMAX, DABS(X(I)))
   DO 110 J=1,M
110 A(J,I)=A1(J,I)
   DO 120 J=1,M
120 B(J)=B1(J)
130 NTAL=NTAL+1
C
   WRITE(1,1000)NTAL
1000 FORMAT(G15.8)
C
   IF (NTAL.GT.MAXIT) GOTO 140
C   TEST OF CONVERGENCE CRITERION
   IF (DX.LE.EPS*XMAX) GOTO 140

```

C

```

      GOTO 30
139 EPS=0.
140 B1(1)=NTAL
      RETURN
      END
      SUBROUTINE MA19W (N,M,A,IA,B,DX,EPS,X,RES,IREF,NURES,NUIREF)
      IMPLICIT REAL*8 (A-H,O-Z), INTEGER*4 (I-N)
      DIMENSION A(IA,N),B(M),X(N),RES(NURES)
      INTEGER*4 IREF(NUIREF)
      NURES=(N+1)*(N+5)+M
      NUIREF=5*(N+1)+M
      N1=N+1
      N2=N+2
      NURHO=MAX0(M,3*N1)+1
      NH=1+NURHO
      CALL MA19BW(N,M,A,IA,B,DX,EPS,X,RES(NH),N1,RES(1),IREF,NURHO,NUIREF,
1F,N2)
      RETURN
      END
      SUBROUTINE MA19BW(N,M,A,IA,B,DX,EPSH,X,H,N1,RHO,IREF,NURHO,NUIREF,
1N2)
      IMPLICIT REAL*8 (A-H,O-Z), INTEGER*4 (I-N)
      REAL*8 A(IA,N),B(M),X(N),H(N1,N2),RHO(NURHO),LAM
      INTEGER*4 IREF(NUIREF)
      LOGICAL GAMCH
      IF ((DX.LT.0D0).OR.(EPSH.LT.0D0)) RETURN
      IF ((N.LT.1).OR.(M.LT.1)) RETURN

```

C

```

      NN2=N+N2
      NN3=NN2+N1
      LREF=NN3+N1
      LBND=LREF+M
      M1=M+1
      SI4N=1./(4.*N)
      NTAL=0

```

C

FIND EQUATION 10 WHICH GOES INTO THE FIRST REFERENCE

```

      C=-1.
      DO 51 J=1,M
      IREF(LREF+J)=0
      IF(DABS(B(J)).LT.C) GOTO 51
      C=DABS(B(J))
      I0=J

```

51 CONTINUE

C

INITIALIZE REFERENCE ARRAY:

```

      S=0D0
      T=B(I0)
      XM=M
      DO 52 I=1,N
      D=A(I0,I)
      S=S+DABS(D)
      XM=XM+1D0
      IF (D.EQ.0D0) D=-1.
      IREF(I)=DSIGN(XM,-D*T)

```

52 IREF(LBND+I)=IREF(I)

```

      XM=I0
      IREF(N1)=DSIGN(XM,T)
      IREF(LREF+I0)=IREF(N1)

```

C

INITIALIZE DH,DC, AND GAM

```

      IF ((DX*S).GT.C) GOTO 53

```

```

C
    GAM=DX
    DG=GAM
    D1=C-DX*S
    GOTO 54
53  DG=C/S
    DH=0.
    GAM=DG
C    FIND VECTOR X
54  DO 55 I=1,N
    XM=IREF(I)
55  X(I)=DSIGN(DG,XM)
C    FIND MATRIX H
    S=1./(S+1.)
    H(N1,N1)=S
    IREF(NN2+N1)=0
    DO 58 I=1,N
    XM=-IREF(I)
    H(N1,I)=DSIGN(S,XM)
    T=DABS(A(I0,I))*S
    H(I,N1)=T
    DO 56 J=1,N
    XM=-IREF(J)
56  H(I,J)=DSIGN(T,XM)
    IF (T.GT.0.) GOTO 57
    IREF(NN2+I)=1
    H(I,N2)=1.
    GOTO 58
57  IREF(NN2+I)=0
58  H(I,I)=ISIGN(1,IREF(I))+H(I,I)
C    INITIALIZE SOME CONSTANTS
    RSIG=1D0-S
    DGH=DG
    DH1=DH
    ETA=GAM
    ERRX=0D0
    GANCH=.TRUE.
    NBIN=N
    GOTO 640
C    ITERATIVE LOOP STARTS HERE
C    FIND VECTOR RHO
500 DGH=DG
    IF (IOS.LT.0) GOTO 503
    DO 502 I=1,N1
    S=-H(I,N1)
    DO 501 J=1,N
501  S=S-H(I,J)*A(I0,J)
502  RHO(I)=S
    GOTO 520
503 DO 505 I=1,N1
    S=-H(I,N1)
    DO 504 J=1,N
504  S=S+H(I,J)*A(I0,J)
505  RHO(I)=S
    GOTO 520
C    BOUNDS VIOLATED
510 IO=N+J0
    DHH=DH
    IOS=DSIGN(1.D0,X(J0))
    IF (IOS.LT.0) GOTO 512

```

C

```

DO 511 I=1,N1
511 RHO(I)=-H(I,J0)-H(I,N1)
COTO 520
512 DO 513 I=1,N1
513 RHO(I)=H(I,J0)-H(I,N1)
C    FIND EQUATION L0 WHICH LEAVES THE REFERENCE
C    FIND -H(I,N1)/RHO(I) FOR NEGATIVE VALUES OF RHO(I)
520 LB=0
RTAU=0D0
IF (I0.GT.M) RTAU=1.
DO 522 I=1,N1
IF (RHO(I).GE.0.D0) COTO 521
LB=LB+1
IREF(N1+LB)=I
KK=N1
IF (IREF(NN2+1).GT.0) KK=N2
RHO(N1+I)=-H(I,KK)/RHO(I)
521 IF (IADS(IREF(I)).LE.M) COTO 522
RTAU=RTAU+RHO(I)
522 CONTINUE
C    FIND THE COEFFICIENTS IN THE RATIONAL EXPRESSION
C    (TT+LAM*SS)/(RSIG+LAM*RTAU)
DC2=DC*2.
IF (DH.GT.0D0) COTO 523
TT=DC*RSIG
SS=DC*RTAU+DHH+DGH-DC
NL=1
COTO 524
523 RSIG=1.-RSIG
RTAU=-RTAU
TT=DH*RSIG
SS=DH*RTAU+DHH-DH+DGH-DC
NL=2
524 SMAX=0.
L=1
525 LA=L
C    FIND MINIMUM VALUE OF -H(I,N1)/RHO(I)
L0=IREF(N1+LA)
LAM=RHO(N1+L0)
ILAM=IREF(NN2+L0)
IF (LA.EQ.LB) COTO 527
L=LA+1
LN=LA
DO 526 I=L,LB
IN=IREF(N1+I)
S=RHO(N1+IN)
IS=IREF(NN2+IN)
IF (((IS.EQ.ILAM).AND.(S.GE.LAM)).OR.(IS.LT.ILAM)) COTO 526
L0=IN
LAM=S
ILAM=IS
LN=I
526 CONTINUE
C    REORDER
IREF(N1+LN)=IREF(N1+LA)
IREF(N1+LA)=L0
C    FIND MAX(-RHO(I))
527 SMAX=DMAX1(SMAX,-RHO(L0))
C    REVISE THE COEFFICIENTS OF THE RATIONAL EXPRESSION

```


C

```

      ML=NL
      IF (IABS(IREF(L0)).GT.M) ML=NL+2
      GOTO (530,529,529,528),ML
528  TT=TT+DC2*H(L0,N1)
      SS=SS+DC2*RHO(L0)
      GOTO 530
529  RSIG=RSIG-2.*H(L0,N1)
      RTAU=RTAU-2.*RHO(L0)
530  IF ((RSIG*SS.GT.RTAU*TT).AND.(LA.LT.LB)) GOTO 525
C    TEST IF -RHO(L0) IS TOO SMALL
      SMAX=SMAX/4.
540  IF(-RHO(L0).GE.SMAX) GOTO 550
      LA=LA-1
      L0=IREF(N1+LA)
      GOTO 540
C    UPDATE REFERENCE ARRAYS
550  IREF(LREF+10)=ISIGN(1,10S)
      IREF(LREF+IABS(IREF(L0)))=0
      IF (IABS(IREF(L0)).GT.M) NBIN=NBIN-1
      IF (10.GT.M) NBIN=NBIN+1
      IREF(L0)=ISIGN(10,10S)
C    UPDATE MATRIX H
      RHOL0=RHO(L0)
      DO 560 J=1,N1
560  H(L0,J)=-H(L0,J)/RHOL0
      DO 562 I=1,N1
      IF (I.EQ.L0) GOTO 562
      S=RHO(I)
      DO 561 J=1,N1
561  H(I,J)= S*H(L0,J)+H(I,J)
562  CONTINUE
C    IF ANY SIGNS HAVE BEEN CHANGED, UPDATE H
      RHON1=1D0
      DO 569 I=1,N1
569  RHO(N1+I)=1.
      IF (LA.LE.1) GOTO 590
      K=LA-1
      DO 570 I=1,N1
570  RHO(NN2+I)=0D0
      DO 571 I=1,K
571  RHO(IREF(N1+I)+N1)=-1.
      DO 575 I=1,N1
      IF (RHO(N1+I).GT.0.) GOTO 572
      DO 572 J=1,N1
572  RHO(NN2+J)=RHO(NN2+J)-H(I,J)
      GOTO 575
573  DO 574 J=1,N1
574  RHO(NN2+J)=RHO(NN2+J)+H(I,J)
575  CONTINUE
      RHON1=RHO(NN2+N1)
      DO 576 J=1,N1
576  H(J,N1)=H(J,N1)/RHON1
      DO 578 I=1,N
      S=RHO(NN2+I)
      DO 577 J=1,N1
577  H(J,I)=-S*H(J,N1)+H(J,I)
578  CONTINUE
C    CHANCE SIGNS IN SOME ROWS OF MATRIX H
      DO 580 L=1,K

```

C

```

      I=IREF(N1+L)
      J=-IREF(I)
      IREF(I)=J
      IREF(LREF+IABS(J))=J
      DO 530 J=1,N1
520  H(I,J)=-H(I,J)
C      UPDATE THE LAST COLUMN OF H IN CASE OF DEGENERACIES
590  IF (IREF(NN2+L0).EQ.0) GOTO 600
      H(L0,N2)=-H(L0,N2)/(RHO(L0)*RHON1)
      DO 595 I=1,N1
      IF ((IREF(NN2+I).EQ.0).OR.(I.EQ.L0)) GOTO 593
      IF (IREF(NN2+I)-IREF(NN2+L0)) 591,593,592
591  H(I,N2)=H(I,N2)/RHON1
      GOTO 595
592  H(I,N2)=DABS(H(L0,N2)*RHO(I))
      IREF(NN2+I)=IREF(NN2+L0)
      GOTO 595
593  C=H(I,N2)/RHON1+H(L0,N2)*RHO(I)
      IF ((LA.GT.1).AND.(RHO(N1+I).LT.0D0)) C=-C
      IF (C.GT.0) GOTO 594
      H(I,N2)=1.D0
      IREF(NN2+I)=IREF(NN2+I)+1
      GOTO 595
594  H(I,N2)=C
595  CONTINUE
      GOTO 610
C      TEST FOR DEGENERACIES
600  DO 601 I=1,N1
      IREF(NN2+I)=0
      IF (H(I,N1).GT.0D0) GOTO 601
      IREF(NN2+I)=1
      H(I,N1)=0D0
      H(I,N2)=1D0
601  CONTINUE
C      UPDATE GAM
610  GAMCH=.FALSE.
      IF ((NBIN.EQ.0).OR.((GAMM.LT.2.*GAM).AND.(GAMM.LT.DX))) GOTO 620
      IF (GAM.LT.GAMM) GAMCH=.TRUE.
      GAM=GAMM
C      UPDATE DH AND DG
620  S=0.
      RSIG=0D0
      DO 622 I=1,N1
      K=IABS(IREF(I))
      IF (K.GT.M) GOTO 621
      RES(I)=B(K)*ISIGN(1,IREF(I))
      S=S+RES(I)*RHO(I)
      GOTO 622
621  RSIG=RSIG+H(I,N1)
622  CONTINUE
      IF (RSIG.NE.0D0) GOTO 623
      DH=DABS(S)
      DG=GAM
      GOTO 630
623  DC=DMIN1(GAM,DABS(S)/RSIG)
      DH=0.
      IF (DC.EQ.GAM) DH=DABS(S-DG*RSIG)/(1D0-RSIG)
C      CALCULATE PARAMETER VALUES
630  DGH=0D0

```

C

```

ERRX=0.D0
DO 632 I=1,N1
IF (IABS(IREF(I)).GT.M) GOTO 631
RHO(I)=DH-RHO(I)
COTO 632
631 RHO(I)=DG
632 CONTINUE
DO 636 I=1,N
S=H(I,I)*RHO(I)
DO 633 J=2,N1
633 S=S+H(J,I)*RHO(J)
IF (IREF(LBND+I).EQ.0) GOTO 634
T=S
XM=IREF(LBND+I)
S=DSIGN(DG,XM)
ERRX=DMAX1(ERRX,DABS(S-T))
634 X(I)=S
IF (DABS(S).LE.DGH) GOTO 636
DGH=DABS(S)
J0=I
636 CONTINUE
NTAL=NTAL+1
C CALCULATE GAMM
640 GAMM=DMIN1(DMAX1(5.*DG,GAM),DX)
C FIND EQUATION 10 WHICH GOES INTO THE REFERENCE
DEH=OD0
T=DH
T1=DH
DO 652 I=1,M
S=B(I)
DO 650 J=1,N
650 S=S+A(I,J)*X(J)
RHO(I)=S
IF (IREF(LREF+I).NE.0) GOTO 651
IF (DABS(S).LE.DHH) GOTO 652
DHH=DABS(S)
IOS=DSIGN(1.D0,S)
I0=I
GOTO 652
651 T=DMAX1(T,DABS(S))
T1=DMIN1(T1,DABS(S))
652 CONTINUE
C CALCULATE DH1 AND ETA
ETA=DMAX1(GAMM,DSIGN(ETA+ERRX,DH1-T))
IF (.NOT.CANCH) T=DMAX1(T,DH1+(T-T1))
DH1=DMAX1(T,DH+EPSH)
C TEST IF CONSTRAINTS ARE VIOLATED
IF (DGH.GT.ETA) GOTO 510
C TEST OF CONVERGENCE CRITERION
IF (DHH.GT.DH1) GOTO 500
IF ((GAM.LT.DX).AND.(DHH+DH.GT.(DH1-DH)*2.)).AND.(NBIN.GT.0)) GOTO
1610
RHO(M1)=DH
IREF(N2)=NBIN
IREF(N+3)=NTAL
RETURN
END

```

7.7 One Dimensional Graphics Plotting Program

C **** TWO DIMENSIONAL PLOTTING PROGRAM ****

C **** TWO DIMENSIONAL PLOTTING PROGRAM ****

```

C      COMPILE   FTM PLOT2D          FTM PLOT2D -64V
C      LOAD      LOAD                SEG
C
C      LO R_PLOT2D          LOAD *PLOT2D
C      LI HGP1F_NONSHARED   LO R_PLOT2D
C      LI          LI          LI HGP1FV_NONSHARED
C      SA *PLOT2D          SA
C      Q              Q
C      RUN:      R *PLOT2D          SEG *PLOT2D
C              (USES A SET-UP DATA FILE CALLED D_PLT2D
C              AND ASKS FOR A FILE NAME OF DATA POINTS )

```

```

C      REAL XX(620),YY(620,8)
CC     REAL XX(2048),YY(2048,8)
C      DIMENSION IFORM(40),IFILE(16)
C      CALL SRCH$$(INTS(1),'D_PLT2D',INTS(7),INTS(5),ITYPE,ICODE)

```

```

C
C 10      CALL TNOUA('DATA FILE NAME = ',INTS(17))
C      READ(1,100)IFILE
100      FORMAT(16A2)
101      FORMAT(2I3,4G15.8)
CC101    FORMAT(2I4,4G15.8)
C      CALL SRCH$$(INTS(1),IFILE,INTS(32),INTS(4),ITYPE,ICODE)
C      IF(ICODE.EQ.INTS(0))GO TO 18
C      WRITE(1,106)IFILE
106      FORMAT('DATA FILE ',16A2)
C      CALL TNOU(' WAS NOT FOUND. PLEASE RE-ENTER NAME.',INTS(38))
C      GO TO 10
120      WRITE(1,107)IFILE.
107      FORMAT('THERE IS A FORMAT PROBLEM WITH ',16A2)
C      CALL TNOU('PLEASE GIVE A NEW DATA FILE NAME',INTS(32))
C      CALL SRCH$$(INTS(4),INTS(0),INTS(0),INTS(4),ITYPE,ICODE)
C      GO TO 10
18      READ(8,101,ERR=120)NR,NC,XMIN,XMAX,YMIN,YMAX
C      IF(NR.LT.INTS(1).OR.NC.LT.INTS(1))GO TO 90
C      IF(NR.EQ.INTS(1).AND.NC.EQ.INTS(1))GO TO 500
C      IF(NR.GT.INTS(192).OR.NC.GT.INTS(192))GO TO 90
C      READ(8,104)IFORM(1)
C      IF(IFORM(1)/256.NE.'( '/256)GO TO 16
C      REWIND 8
C      READ(8,101)NR,NC,XMIN,XMAX,YMIN,YMAX
C      READ(8,104)IFORM
104      FORMAT(40A2)
C      IF(NC.GT.8)NC=8
C      READ(8,IFORM,END=300,ERR=16)((YY(I,J),J=1,NC),I=1,NR)
C      GO TO 17
16      REWIND 8
C      READ(8,101)NR,NC,XMIN,XMAX,YMIN,YMAX
C      IF(NC.GT.8)NC=8
C      READ(8,END=300)((YY(I,J),J=1,NC),I=1,NR)
17      CALL SRCH$$(INTS(4),INTS(0),INTS(0),INTS(4),ITYPE,ICODE)
C      NRMAX=NR
C      NCMAX=NC

```

```

C
C      DO 1 I=1,NR

```

C **** TWO DIMENSIONAL PLOTTING PROGRAM ****

```

1      XX(I)=XMIN+(I-1.)/(NR-1.)*(XMAX-XMIN)
      READ(9,400)NORMAL
400  FORMAT(I1)
      IF(NORMAL.EQ.0)GO TO 6
      YMAX=YY(1,1)
      DO 4 J=1,NC
      DO 4 I=1,NR
      IF(YMAX.LT.YY(I,J))YMAX=YY(I,J)
4      CONTINUE
      DO 5 J=1,NC
      DO 5 I=1,NR
      YY(I,J)=YY(I,J)/YMAX
5      CONTINUE
6      CONTINUE
      CALL GNGRAF(XX,YY(1,1),NR,1,1)
      IF(NC.EQ.1)GO TO 300
      DO 3 J=2,NC
      JJ=J
      IF(JJ.GT.4)JJ=J-4
      CALL GNGRAF(XX,YY(1,J),NR,JJ,0)
3      CONTINUE
300   CALL SRCHSS(INTS(4),INTS(0),INTS(0),INTS(5),ITYPE,ICODE)
      CALL EXIT
      END

C      SUBROUTINE GNGRAF(X,Y,N,II,JJ)
C
C      GENERAL GRAPHING ROUTINE OF X,Y LINEAR DATA
C      WHERE X IS LIST OF X VALUES
C      Y IS LIST OF Y VALUES
C      N IS LENGTH OF LISTS
C
C      TO BE BOUND WITH IOBLOC BLOCK DATA SUBPROGRAM
C
C      DIMENSION X(1),Y(1),NX(10),NY(10),ITI(15),NHOL(30)
C      DATA IHR,IHP,IHQ,IHM,IHG/IHR,IHP,IHQ,IHM,IHG/
C
C      MDVIN=9
C      MDVOUT=1
C      IF(JJ.NE.1)GO TO 200
C      CALL GRESET
C      CALL SET2D
C
C      ERASE SCREEN
5      CALL ENTGRA
      CALL ERASS
      CALL EXITGR
C      WRITE (MDVOUT,906)
C      SET ORIGIN
200   XORG=-4.
      YORG=-4.
      CALL ENTGRA
      CALL INIT(XORG,YORG)
      CALL EXITGR
      IF(JJ.NE.1)GO TO 100
C      INPUT AXES LENGTH
10      WRITE (MDVOUT,900)
      READ (MDVIN,800,ERR=10) XLN
20      WRITE (MDVOUT,901)

```

C **** TWO DIMENSIONAL PLOTTING PROGRAM ****

```
      READ (NDVIN,800,ERR=20) YLN
      IF (XLN.LE.2.0 .OR. XLN.GE.11.) XLN=5.
      IF (YLN.LE.2.0 .OR. YLN.GE.11.) YLN=5.
C SET TITLE COORDINATES
      XTI=0.
      YTI=YLN+.3
C SCALE THE DATA
      WRITE (NDVOUT,934)
934   FORMAT(42H ENTER 0-LINEAR, 1- XLOG, 2-YLOG, 3-LOGLOG)
      READ (NDVIN,935) ITLOG
935   FORMAT(I1)
      IF (ITLOG.LT.0 .OR. ITLOG.GT.3) ITLOG=0
      LGY=IAND(ITLOG,2)
      LGX=IAND(ITLOG,1)
      IF (LGX.EQ.0) CALL LINSAX(X,N,XLN,0)
      IF (LGX.EQ.1) CALL LOGSAX(X,N,XLN,0)
      IF (LCY.EQ.2) CALL LOGSAX(Y,N,YLN,1)
      IF (LCY.EQ.0) CALL LINSAX(Y,N,YLN,1)
C REQUEST GRID
30   WRITE (NDVOUT,905)
      READ (NDVIN,803,ERR=30) IGRD
C REQUEST AXES TITLES
40   WRITE (NDVOUT,902)
      READ (NDVIN,802,ERR=40) (NHOL(I),I=1,20)
C
C PACK STRING INTO NX
      DO 90 I=1,20
      CALL PUTCHR (NHOL(I),NX,I)
90   CONTINUE
C
C GET YAX TITLE. PACK INTO NY
45   WRITE (NDVOUT,903)
      READ (NDVIN,802,ERR=45) (NHOL(I),I=1,20)
      DO 91 I=1,20
      CALL PUTCHR (NHOL(I),NY,I)
91   CONTINUE
C
C REQUEST GRAF TITLE
50   WRITE (NDVOUT,904)
      READ (NDVIN,801,ERR=50) (NHOL(I),I=1,30)
      DO 92 I=1,30
      CALL PUTCHR (NHOL(I),ITI,I)
92   CONTINUE
C
C DRAW GRAPH
      CALL ENTGRA
54   CALL INIT(XORG,YORG)
      CALL AXPRES(3)
55   CALL ERASS
      IF (LGX.EQ.0) CALL XLINAX(NX,20,XLN)
      IF (LCY.EQ.0) CALL YLINAX(NY,20,YLN)
      IF (LCY.EQ.2) CALL YLOGAX(NY,20,YLN)
      IF (LGX.EQ.1) CALL XLOGAX(NX,20,XLN)
C
      IF (IGRD.NE.IHC) GO TO 60
C      CALL CRID (XLN,YLN,1.,1.,4)
      CALL AXGRID(1.,1.,4)
60   CALL PLOT (XTI,YTI,0)
C
```

C **** TWO DIMENSIONAL PLOTTING PROGRAM ****

```
      CALL SLTSIZ(5)
      CALL TEXT(ITI,30)
      CALL EXITGR
100    CONTINUE
      CALL ENTGRA
      CALL DATLIN (X,Y,N,11,0,700)
      CALL EXITGR

C
C IF AN R IS TYPED, THEN RESTART
C IF A P IS TYPED, THEN REPOSITION TITLE
C IF AN M IS TYPED, THEN MOVE ENTIRE GRAPH
C
      READ(MDVIN,957)INP
957    FORMAT(A1)
      IF (INP.EQ.IHR) GO TO 5
      IF (INP.EQ.IHM) GO TO 70
      IF (INP.NE.IHP) RETURN
C GET NEW TITLE POSITION
      CALL ENTGRA
C      IWAIT=1
      CALL ENBDEV(1,1,1,1,0)
      CALL GETXY(1,IREADY,XTI,YTI)
      CALL DISDEV(1,0)

C
C      CALL ENTGRA
      XTI=XTI-XORG
      YTI=YTI-YORG
      GO TO 53
C REPOSITION GRAPH
70    CALL ENTGRA
      CALL ENBDEV(1,1,1,1,0)
      CALL GETXY(1,IREADY,XORG,YORG)
      CALL DISDEV(1,0)
      GO TO 54
C FORMAT STATEMENTS
900    FORMAT (49H INPUT 2.<XAXIS LENGTH<11.(1 DEFAULT 5. INCHES):)
901    FORMAT (49H INPUT 2.<YAXIS LENGTH<11.(1 DEFAULT 5. INCHES):)
902    FORMAT (38H INPUT XAXIS TITLE UP TO 20 CHARACTERS,/,2H :)
903    FORMAT (38H INPUT YAXIS TITLE UP TO 20 CHARACTERS,/,2H :)
904    FORMAT (41H INPUT TITLE OF GRAPH UP TO 30 CHARACTERS,/,2H :)
905    FORMAT (29H IF YOU WANT A GRID TYPE G:)
906    FORMAT (3X,24H HELLO, IN GENERAL GRAPH/24H JUST FILL IN THE BLANKS/
X47H AND DONT FORGET DECIMAL POINTS IN AXIS LENGTHS/
X40H ILL WAIT FOR YOU WHEN IM DONE DRAWING /
X39H AT THAT TIME TYPE ONE OF THE FOLLOWING//
X7H R-REDO/17H P-POSITION TITLE/13H M-MOVE GRAPH/11H ELSE-QUIT//)
800    FORMAT (F2.0)
801    FORMAT (30A1)
802    FORMAT (20A1)
803    FORMAT (A1)
      END

C
C
C
C      SUBROUTINE QCALE (AMN,AMX,AXLR,IWHO)
      COMMON /PLT2C/ITYPE
      COMMON /PLT2E/XIN,YIN,XMX,YMX
C      QCALE ACTUAL LINEAR SCALE ROUTINE
```



```

C **** TWO DIMENSIONAL PLOTTING PROGRAM ****

C  PARAMETERS ARE:
C  AMN - MIN VAL OF USER DATA
C  AMX - MAX VALUE
C  AXLH - SEE ROUTINE SCALE
C  IWHO - 0-XAXIS ELSE Y
C
C  ST IS LIST OF CANONICAL TIC INTERVALS THAT CAN BE USED
C  DIMENSION ST(13)
C  DATA ST/1.,2.,2.5,4.,5.,8.,10.,20./
C  DATA ST/1.,1.25,1.5,2.,2.5,3.,3.33333,4.,5.,8.,10.,15.,20./
C
C  STORE MIN AND MAX
C  AMN1=AMN
C  UPIN=(AMX-AMN1)/AXLH
C  IF (UPIN.EQ.0.) GO TO 11
C
C  FIND SCALE MAGNITUDE
C  ASNQ=ALOG10(UPIN)
C  N=IFIX(ASNQ)
C  FACT=10.**N
C  IF (UPIN.LT.1.) FACT=FACT/10.
C
C  FIND INCREMENT LARGER THAN UPIN(UNITS/INCH)
C  DO 10 K=1,13
C  SF=FACT*ST(K)
C  IF (SF.GE.UPIN) GO TO 20
C  10 CONTINUE
C  GET INTEGER MULTIPLIER FOR MINIMUM
C  20 N=AMN1/SF
C  IF (N.LT.0) N=N-1
C  FPN=FLOAT(N)
C  IF (FPN*SF.GT.AMN1) N=N-1
C  FPN=FLOAT(N)
C  VLO=FPN*SF
C  30 IF (VLO+SF*AXLH .GE. (AMX-AMX*.0001)) GO TO 40
C
C  IF NOT IN LIMITS TRY NEXT SIZE
C  K=K+1
C  SF=ST(K)*FACT
C  GO TO 20
C  11 IF (AMN1.LT.0.) AMN1=AMN1-1.
C  N=IFIX(AMN1)
C  VLO=FLOAT(N)
C  SF=1.
C  40 CALL YOURSC(SF,VLO,IWHO)
C
C  SET SCALE TYPE INTERNAL PARAMETER
C  IF (IWHO.EQ.0) ITYPE=IAND(ITYPE,2)
C  IF (IWHO.EQ.0) XMN=AMN
C  IF (IWHO.EQ.0) XMX=AMX
C  IF (IWHO.NE.0) YMN=AMN
C  IF (IWHO.NE.0) YMX=AMX
C  IF (IWHO.NE.0) ITYPE=IAND(ITYPE,1)
C  RETURN
C  END
C
C
C

```

7.8 Two Dimensional Surface Plotting Program

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

C      COMPILE: FTM PLOT3D -SPO -64V
C      LOAD:      SEG
C                  LO #PLOT3D
C                  LO B_PLOT3D
C                  LI HGP1MV
C                  LI
C                  SA
C                  Q
SINSERT SYSCOM>KEYS.F
SINSERT SYSCOM>ERRD.F
      INTEGER*2 IFILE,IFORM,IVAL,LOG,NXL,NYL,NZL
      INTEGER*2 ITYPE,IBIN,ICODE,NR,NC,I,J,NRMAX,NCMAX
      INTEGER*2 KODE,LINDR,MGN,NCHNG,IVAR
      REAL*4 PITCH,SCLE,SDIST,SIZE,YAW,T,SCL,RANGE,S,VALUE,DE,DV
      REAL*4 XMIN,XMAX,DELT,A,AA,YMIN,YMAX,ZMIN,ZMAX,SS,H,V,X,Y,Z,XP
      REAL*4 H1,H2,H3,H4,H5,H6,H7,H8,V1,V2,V3,V4,V5,V6,V7,V8
      DIMENSION A(192,192),AA(192,192),IFILE(16),IFORM(40)
      DIMENSION H(10),V(10),X(2),Y(2),Z(2),XP(8)
      COMMON ANGA,ANCB,HV,D,SH,SV
      COMMON SL,SM,SN,CX,CY,CZ,QX,QY,QZ,SD
      COMMON/AREA1/ A
      COMMON/AREA2/ AA
10     CALL TNOUA('DATA FILE NAME = ',INTS(17))
      READ(1,100)IFILE
100    FORMAT(16A2)
101    FORMAT(2I3,4G15.8)
      CALL SRCH33(INTS(1),IFILE,INTS(32),INTS(5),ITYPE,ICODE)
      IF(ICODE.EQ.INTS(0))GO TO 18
      WRITE(1,106)IFILE
106    FORMAT('DATA FILE ',16A2)
      CALL TNOU(' WAS NOT FOUND. PLEASE RE-ENTER NAME.',INTS(38))
      GO TO 10
120    WRITE(1,107)IFILE
107    FORMAT('THERE IS A FORMAT PROBLEM WITH ',16A2)
      CALL TNOU('PLEASE GIVE A NEW DATA FILE NAME',INTS(32))
      CALL SRCH33(INTS(4),INTS(0),INTS(0),INTS(5),ITYPE,ICODE)
      GO TO 10
18     READ(9,101,ERR=120)NR,NC,XMIN,XMAX,YMIN,YMAX
      IF(NR.LT.INTS(1).OR.NC.LT.INTS(1))GO TO 90
      IF(NR.EQ.INTS(1).AND.NC.EQ.INTS(1))GO TO 500
      IF(NR.GT.INTS(192).OR.NC.GT.INTS(192))GO TO 90
      READ(9,104)IFORM(1)
      IF(IFORM(1)/256.NE.'( '/256)GO TO 16
      REWIND 9
      READ(9,101)NR,NC,XMIN,XMAX,YMIN,YMAX
      READ(9,104)IFORM
104    FORMAT(40A2)
      READ(9,IFORM,END=300,ERR=16)((AA(I,J),J=1,NC),I=1,NR)
      GO TO 17
16     REWIND 9
      READ(9,101)NR,NC,XMIN,XMAX,YMIN,YMAX
      READ(9,END=300)((AA(I,J),J=1,NC),I=1,NR)
17     CALL SRCH33(INTS(4),INTS(0),INTS(0),INTS(5),ITYPE,ICODE)
      NRMAX=NR
      NCMAX=NC
24     CALL TNOU('0 FOR LINEAR PLOT',INTS(17))
      CALL TNOU('1 FOR LOG PLOT',INTS(14))
12     CALL TNOUA('LOG = ',INTS(6))

```

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

READ(1,102,ERR=24)LOG
IF(LOG.LT.0.OR.LOG.GT.1)GO TO 12
DO 19 I=1,NR
DO 19 J=1,NC
A(I,J)=AA(I,J)
IF(LOG.EQ.0)GO TO 19
IF(A(I,J).EQ.0.)A(I,J)=1.E-37
A(I,J)=ALOG10(ABS(A(I,J)))
19 CONTINUE
20 CALL TNOU('0 FOR HIDDEN LINES',INTS(18))
CALL TNOU('1 FOR ALL LINES ',INTS(16))
23 CALL TNOU('KODE = ',INTS(7))
READ(1,102,ERR=20)KODE
102 FORMAT(I2)
IF(KODE.LT.0.OR.KODE.GT.1)GO TO 23
25 CALL TNOU('1-FOR LINES IN X DIRECT. ONLY',INTS(29))
CALL TNOU('2-FOR LINES IN Y DIRECT. ONLY',INTS(29))
CALL TNOU('3-FOR LINES IN BOTH DIRECTIONS',INTS(29))
27 CALL TNOU('LINDR = ',INTS(8))
READ(1,102,ERR=25)LINDR
IF(LINDR.LT.0.OR.LINDR.GT.3)GO TO 27
30 CALL TNOU('PITCH = ',INTS(8))
READ(1,103,ERR=30)PITCH
103 FORMAT(F5.1)
IF(ABS(PITCH).GT.360.)GO TO 30
35 CALL TNOU('SCALE = ',INTS(8))
READ(1,103,ERR=35)SCLE
IF(SCLE.LT.0.0.OR.SCLE.GT.2.0)GO TO 35
40 CALL TNOU('DISTANCE = ',INTS(11))
READ(1,103,ERR=40)SDIST
IF(SDIST.LT.1.0.OR.SDIST.GT.150.)GO TO 40
45 CALL TNOU('SIZE = ',INTS(7))
READ(1,103,ERR=45)SIZE
IF(SIZE.LT.5.0.OR.SIZE.GT.12.)GO TO 45
50 CALL TNOU('YAW = ',INTS(6))
READ(1,103,ERR=50)YAW
IF(ABS(YAW).GT.360.)GO TO 50
13 CALL TNOU('0 FOR NO BOX OR AXES',INTS(20))
CALL TNOU('1 FOR REFERENCE BOX',INTS(19))
CALL TNOU('2 FOR BOX AROUND PLOT',INTS(21))
CALL TNOU('3 FOR X, Y & Z AXES',INTS(19))
CALL TNOU('5 FOR LABELED AXES & BOX',INTS(23))
14 CALL TNOU('MCN = ',INTS(6))
READ(1,102,ERR=13)MCN
IF(MCN.LT.0.OR.MCN.GT.5)GO TO 14
CALL GRESET

C
C
C
55 CALL ENTGRA
CALL ERSALL
CALL OPENVU(1)
CALL STTLP(-4.6,-4.8)
CALL STGES(1)
CALL CLOSUV
CALL EXITCR
WRITE(1,200)NR,NC,SDIST,YAW,PITCH,LINDR,KODE,SIZE,SCLE,LOG,MCN
200 FORMAT(' AN ARRAY OF ',I3,' BY ',I3,' IS DRAWN' /

```

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

1'  DISTANCE='F6.1/'  YAW      ='F6.1/'  PITCH  ='F6.1/
2'  LINDR    ='13/'    KODE     ='13/'    SIZE   ='F6.1/
3'  SCALE    ='F6.2/'  LOG      ='13/'    NGN    ='13)

```

C
C
C

```

      CALL ENTGRA
      CALL DSPICT(1)
      CALL OPNV(1,1)
      CALL SLTRZ(INTS(0))
      DO 56 I=1,2
        X(I)=0.0
        Y(I)=0.0
        Z(I)=0.0
        H(I)=0.0
        V(I)=0.0
        XP(I)=0.0
56      CONTINUE
        DO 57 I=3,8
          XP(I)=0.0
          H(I)=0.0
          V(I)=0.0
57      CONTINUE
        DO 58 I=9,10
          H(I)=0.0
          V(I)=0.0
58      CONTINUE
      C
      ANGA=(YAW+270.)*0.0174532
      ANGB=PITCH*0.0174532
      HV=SIZE
      C
      DIRECTION COMPONENTS OF THE EYE
      SL=-COS(ANGA)*COS(ANGB)
      SM=-SIN(ANGA)*COS(ANGB)
      SN=-SIN(ANGB)
      C
      IF(ABS(SN).EQ.1.0)GO TO 11
      C
      SD=1.0/SQRT(1.-SN*SN)
      X(1)=1
      X(2)=NR
      Y(1)=1
      Y(2)=NC
      T=MAX0(NR,NC)
      C
      FIND THE DIAGONAL OF THE CUBE
      D=NC**2+NR**2+T**2
      D=SQRT(D)
      SCL=SDIST*D
      C
      FIND COORDINATES OF YOUR EYE
      CX=-SL*SCL
      CY=-SM*SCL
      CZ=-SN*SCL
      C
      COORDINATES OF THE PROJECTION PLANE
      QX=CX+D*SL
      QY=CY+D*SM
      QZ=CZ+D*SN
      C
      FIND MAX AND MIN OF A
      CALL THRE1(A,NR,NC,Z,KODE)
      ZMIN=Z(1)

```

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

      ZMAX=Z(2)/SCLE
      RANGE=(Z(2)-Z(1))
      S=1.
      IF(SCLE.NE.0.)S=T/RANGE*SCLE
      SS=S
C     SCALE THE SURFACE TO MAKE A CUBE
      DO 31 I=1,NR
      DO 31 J=1,NC
      A(I,J)=(A(I,J)-Z(1))*S
31    CONTINUE
      Z(1)=0.0
      Z(2)=T
C
2080  CALL THRE2(X,Y,Z,XP,H,V,KODE)
C
      DO 2130 I=1,8
      H(I)=((XP(I)-QX)*SM-(H(I)-QY)*SL)*SD
      V(I)=(V(I)-QZ)*SD
2130  CONTINUE
C
2190  CALL THRE1(H,8,1,H(9),KODE)
2021  CALL THRE1(V,8,1,V(9),KODE)
C
      IF(MGN.EQ.INTS(0))GO TO 2140
      S=HV
      IF(MGN.EQ.INTS(1))S=1.5
C
      CALL SLINT(INTS(6))
      CALL SLLTYP(INTS(4))
C
      SH=S/(H(10)-H(9))
      SV=S/(V(10)-V(9))
      SH=SIGN(AMIN1(SH,SV),SH)
      SV=SIGN(SH,SV)
      H1=(H(1)-H(9))*SH
      H2=(H(2)-H(9))*SH
      H3=(H(3)-H(9))*SH
      H4=(H(4)-H(9))*SH
      H5=(H(5)-H(9))*SH
      H6=(H(6)-H(9))*SH
      H7=(H(7)-H(9))*SH
      H8=(H(8)-H(9))*SH
      V1=(V(1)-V(9))*SV
      V2=(V(2)-V(9))*SV
      V3=(V(3)-V(9))*SV
      V4=(V(4)-V(9))*SV
      V5=(V(5)-V(9))*SV
      V6=(V(6)-V(9))*SV
      V7=(V(7)-V(9))*SV
      V8=(V(8)-V(9))*SV
      IF(MGN.EQ.INTS(1))CALL MOVE(0.,2.)

```

```

C
C      . 4-----8
C 2-----6 ' |
C |   |   |   |
C |   |   |   |
C | . 3-----7
C 1-----5 '
C

```

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

C      CALL SYMBOL(H1,V1,0.14,'O',0.,1)
C      CALL SYMBOL(H1,V3,0.14,'M',0.,1)
C      CALL SYMBOL(H2,V2,0.14,'Z',0.,1)
C      CALL SYMBOL(H5,V5,0.14,'N',0.,1)
C      CALL MOVE(.03,.05)
C
C      CALL MOVE(H1,V1)
C      CALL DRAW(H2,V2)
C      CALL MOVE(H3,V3)
C      CALL DRAW(H1,V1)
C      CALL DRAW(H5,V5)
C      IF(MGN.EQ.INTS(3))GO TO 2139
C      IF(MGN.EQ.INTS(1).OR.MGN.EQ.INTS(2))GO TO 2138
C      PUT TICK MARKS AND DIMENSIONS ON AXES
C      NXL=IFIX(SIZE*NR/T)
C      IF(NXL.LT.1)NXL=1
C      NYL=IFIX(SIZE*NC/T)
C      IF(NYL.LT.1)NYL=1
C      NZL=IFIX(SIZE)
C      IF(NZL.LT.1)NZL=1
C      DH=(H2-H1)/NZL
C      DV=(V2-V1)/NZL
C      CALL MOVE(H1-.4,V1)
C      CALL NUMBRQ(ZMIN,3,2)
C      CALL MOVE(H1-.08,V1)
C      CALL DRAW(H1,V1)
C      DO 125 I=1,NZL
C      CALL RMOVE(DH-.08,DV)
C      CALL RDRAW(.08,0.)
125  CONTINUE
C      CALL RMOVE(-.4,0.)
C      CALL NUMBRQ(ZMAX,3,2)
C      DH=(H3-H1)/NYL
C      DV=(V3-V1)/NYL
C      CALL MOVE(H1-.25,V1-.2)
C      CALL NUMBRQ(YMIN,3,2)
C      CALL MOVE(H1,V1-.08)
C      CALL DRAW(H1,V1)
C      DO 126 I=1,NYL
C      CALL RMOVE(DH,DV-.08)
C      CALL RDRAW(0.,.08)
126  CONTINUE
C      CALL RMOVE(-.25,-.2)
C      CALL NUMBRQ(YMAX,3,2)
C      DH=(H5-H1)/NXL
C      DV=(V5-V1)/NXL
C      CALL MOVE(H1-.25,V1-.31)
C      CALL NUMBRQ(XMIN,3,2)
C      CALL MOVE(H1,V1)
C      DO 127 I=1,NXL
C      CALL RMOVE(DH,DV-.08)
C      CALL RDRAW(0.,.08)
127  CONTINUE
C      CALL RMOVE(-.25,-.31)
C      CALL NUMBRQ(XMAX,3,2)
C      IF(MGN.NE.INTS(5))GO TO 2139
2138 CALL MOVE(H7,V7)
C      CALL DRAW(H8,V8)
C      CALL DRAW(H4,V4)

```

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

CALL DRAW(H2,V2)
CALL DRAW(H6,V6)
CALL DRAW(H5,V5)
CALL DRAW(H7,V7)
CALL DRAW(H3,V3)
CALL DRAW(H4,V4)
CALL MOVE(H2,V8)
CALL DRAW(H6,V6)

C
2139 IF(MGN.NE.INTS(1))GO TO 2140
CALL MOVE(AINT((H(10)-H(9))*SH+2.),-2.05)

C
2140 CALL SLINT(INTS(9))
CALL SLLTYP(INTS(1))
CALL THREE(X,Y,A,NR,NC,H,V,LINDR,KODE)

C
2150 CONTINUE
CALL CLOSP

C
C
C
CALL EXITGR
62 CALL TNOU(' DO YOU WANT A HARD COPY?',INTS(28))
CALL TNOU(' 0 FOR NO & 1 FOR YES',INTS(24))
61 CALL TNOUA(' HARD COPY = ',INTS(16))
READ(1,101,ERR=61)J
IF(J.LT.INTS(0).OR.J.GT.INTS(1))GO TO 62
IF(J.EQ.INTS(0))GO TO 60
CALL ENTGRA
CALL HDCOPY(1)
CALL EXITGR
60 CALL TNOUA(' NO. OF CHANGES DESIRED = ',INTS(29))
READ(1,102,ERR=60)NCHNG
IF(NCHNG.EQ.0)CALL EXIT

C
C
C
DO 80 I=1,NCHNG
CALL TNOU('KODE IVAR=0',INTS(15))
65 CALL TNOU('NR IVAR=1',INTS(15))
CALL TNOU('NC IVAR=2',INTS(15))
CALL TNOU('DISTANCE IVAR=3',INTS(15))
CALL TNOU('PITCH IVAR=4',INTS(15))
CALL TNOU('LINDR IVAR=5',INTS(15))
CALL TNOU('YAW IVAR=6',INTS(15))
CALL TNOU('SCALE IVAR=7',INTS(15))
CALL TNOU('SIZE IVAR=8',INTS(15))
CALL TNOU('NEW FILE IVAR=9',INTS(15))
CALL TNOU('LOG IVAR=10',INTS(16))
CALL TNOU('MGN IVAR=11',INTS(16))
89 CALL TNOUA(' IVAR = ',INTS(7))
READ(1,105,ERR=65)IVAR
105 FORMAT(I2)
CALL TNOU('ENTER 99 FOR INTEGER VARIABLES OR',INTS(33))
CALL TNOU('999. FOR REAL VARIABLES WHOSE VALUES',INTS(36))
CALL TNOU('ARE TO BE LEFT UNCHANGED.',INTS(25))
IF(IVAR.EQ.0)GO TO 75
GO TO (66,67,68,69,70,71,72,73,74,78,79),IVAR
66 CALL TNOUA('NR = ',INTS(5))

```


C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

        READ(1,101,ERR=66)IVAL
        IF(IVAL.EQ.99)GO TO 89
        NR=IVAL
        IF(NR.GT.NRMAX)GO TO 66
        GO TO 80
67      CALL TNOUA('NC = ',INTS(5))
        READ(1,101,ERR=67)IVAL
        IF(IVAL.EQ.99)GO TO 89
        NC=IVAL
        IF(NC.GT.NCMAX)GO TO 67
        GO TO 80
68      CALL TNOUA('DISTANCE = ',INTS(11))
        READ(1,103,ERR=68)VALUE
        IF(VALUE.EQ.999.)GO TO 89
        SDIST=VALUE
        IF(SDIST.LT.1.0.OR.SDIST.GT.150.)GO TO 68
        GO TO 80
69      CALL TNOUA('PITCH = ',INTS(8))
        READ(1,103,ERR=69)VALUE
        IF(VALUE.EQ.999.)GO TO 89
        PITCH=VALUE
        IF(ABS(PITCH).GT.360.)GO TO 69
        GO TO 80
70      CALL TNOUA('LINDR = ',8)
        READ(1,101,ERR=70)IVAL
        IF(IVAL.EQ.99)GO TO 89
        LINDR=IVAL
        IF(LINDR.LT.1.OR.LINDR.GT.3)GO TO 70
        GO TO 80
71      CALL TNOUA('YAW = ',INTS(6))
        READ(1,103,ERR=71)VALUE
        IF(VALUE.EQ.999.)GO TO 89
        YAW=VALUE
        IF(ABS(YAW).GT.360.)GO TO 71
        GO TO 80
72      CALL TNOUA('SCALE = ',INTS(8))
        READ(1,103,ERR=72)VALUE
        IF(VALUE.EQ.999.)GO TO 89
        SCLE=VALUE
        IF(SCLE.LT.0.0.OR.SCLE.GT.2.0)GO TO 72
        GO TO 80
73      CALL TNOUA('SIZE = ',INTS(7))
        READ(1,103,ERR=73)VALUE
        IF(VALUE.EQ.999.)GO TO 89
        SIZE=VALUE
        IF(SIZE.LT..5.OR.SIZE.GT.12.)GO TO 73
        GO TO 80
74      CALL TNOUA('NEW FILE NAME = ',INTS(16))
        READ(1,100,ERR=74)IFILE
        CALL SRCHSS(INTS(1),IFILE,INTS(32),INTS(5),ITYPE,ICODE)
        IF(ICODE.EQ.0)GO TO 28
        WRITE(1,106)IFILE
        CALL TNOU('WAS NOT FOUND. PLEASE RE-ENTER NAME.',INTS(38))
        GO TO 74
121     WRITE(1,107)IFILE
        CALL TNOU('PLEASE GIVE A NEW DATA FILE NAME',INTS(32))
        CALL SRCHSS(INTS(4),INTS(0),INTS(0),INTS(5),ITYPE,ICODE)
        GO TO 74
23      READ(9,101,ERR=121)NR,NC,XMIN,XMAX,YMIN,YMAX

```

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

IF(NR.GT.INTS(192).OR.NC.GT.INTS(192))GO TO 90
IF(NR.LT.INTS(1).OR.NC.LT.INTS(1))GO TO 90
IF(NR.EQ.INTS(1).AND.NC.EQ.INTS(1))GO TO 500
READ(9,104)IFORM(1)
IF(IFORM(1)/256.NE.'( '/256)GO TO 15
REWIND 9
READ(9,101)NR,NC,XMIN,XMAX,YMIN,YMAX
READ(9,104)IFORM
READ(9,IFORM,END=300,ERR=16)((A(I,J),J=1,NC),I=1,NR)
GO TO 76
15 REWIND 9
READ(9,101)NR,NC,XMIN,XMAX,YMIN,YMAX
READ(9,END=300)((A(I,J),J=1,NC),I=1,NR)
76 CALL SRCHSS(INTS(4),INTS(6),INTS(6),INTS(5),ITYPE,ICODE)
NRMAX=NR
NCMAX=NC
GO TO 80
75 CALL TNOUA('KODE = ',INTS(6))
READ(1,102,ERR=75)IVAL
IF(IVAL.EQ.99)GO TO 89
IF(IVAL.LT.0.OR.IVAL.GT.1)GO TO 75
KODE=IVAL
GO TO 80
78 CALL TNOUA('LOG = ',INTS(6))
READ(1,102,ERR=78)IVAL
IF(IVAL.EQ.99)GO TO 89
IF(IVAL.LT.0.OR.IVAL.GT.1)GO TO 78
LOG=IVAL
GO TO 80
79 CALL TNOUA('MGN = ',INTS(6))
READ(1,102,ERR=79)IVAL
IF(IVAL.EQ.99)GO TO 89
IF(IVAL.LT.0.OR.IVAL.GT.5)GO TO 79
MGN=IVAL
GO TO 80
80 CONTINUE
DO 129 I=1,NR
DO 129 J=1,NC
A(I,J)=AA(I,J)
IF(LOG.EQ.0)GO TO 129
IF(A(I,J).EQ.0.)A(I,J)=1.E-37
A(I,J)=ALOG10(ABS(A(I,J)))
129 CONTINUE
GO TO 55
C
C
C ERROR MESSAGES & ABNORMAL EXIT
C
90 WRITE(1,206)NR,NC
206 FORMAT('ONE OF THE SUBSCRIPTS IS OUT OF RANGE' /
&'NC = ',I3, ' AND NR = ',I3)
GO TO 207
C
C
11 CALL EXITCR
WRITE(1,21)
21 FORMAT(20X,20('*'),//,'ATTEMPTED VERTICAL VIEW. ERROR.')
GO TO 208
C

```


C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

170 CONTINUE
180 CONTINUE
C
190 RETURN
END

C
C
C
C

SUBROUTINE THRE4(X,Y,Z,XP,YP,ZP,KODE)

C
C
C
C

FIND THE LOCATION OF A POINT IN THE ROTATED CUBE.

COMMON ANGA,ANCB,HV,D,SH,SV
COMMON SL,SM,SN,CX,CY,CZ,QX,QY,QZ,SD
REAL*4 ANGA,ANCB,HV,D,SH,SV,SL,SM,SN,CX,CY,CZ,QY,QZ,SD
REAL*4 X,Y,Z,XP,YP,ZP
REAL*4 SK
INTEGER*2 KODE
SK=D/((X-CX)*SL + (Y-CY)*SM + (Z-CZ)*SN)
XP=CX+SK*(X-CX)
YP=CY+SK*(Y-CY)
ZP=CX+SK*(Z-CZ)

C
C
C
C

RETURN
END

C
C
C
C

SUBROUTINE THRE3(X,Y,A,N,M,H,V,K,KODE)

C
C

COMMON ANGA,ANCB,HV,D,SH,SV
COMMON SL,SM,SN,CX,CY,CZ,QX,QY,QZ,SD
REAL*4 ANGA,ANCB,HV,D,SH,SV,SL,SM,SN,CX,CY,CZ,QX,QY,SD
REAL*4 X,Y,A,H,V,END
REAL*4 XI,DI,TO,T,ZP,XP,HH,VV,DD,YJ
INTEGER*2 MM,NN,L,LD,J,I,L11,L22
DIMENSION X(10),Y(10),H(10),V(10),A(192,192)
INTEGER P,Q,P1,P0
INTEGER*2 UP,DOWN,PEN

C
C
C

DRAW THE FIGURE

END=1./128.

C
C
C

USE 1/32 OR 1/64 FOR FINER INTERPOLATION.

UP=INTS(3)
DOWN=INTS(2)
SH=HV/(H(10)-H(9))
SV=HV/(V(10)-V(9))
SH=SIGN(AMIN1(SH,SV),SH)
SV=SIGN(SH,SV)
MM=M
NN=N

C

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

80   IF(K-1)100,120,100
C
100  IF(K-3)1110,120,1110
C
C    DRAW LINES ALONG THE Y AXIS
C
120  CONTINUE
C
      L=0
      LD=1
      DD=0.5*LD
C
140  DO 1060 J=1,M
      Q=0
      YJ=J
160  DO 1030 I=1,NN
      L=L+LD
C
      XI=L
      CALL THRE5(A,XI,YJ,N,M,P,KODE)
      PEN=UP
      IF(P)510,520,530
510  CONTINUE
      IF(Q)540,550,540
520  CONTINUE
      IF(Q)610,1020,610
530  CONTINUE
      IF(Q)540,550,540
540  CONTINUE
C
      PEN=DOWN
      GO TO 170
C
550  CONTINUE
      IF(I.EQ.1)GO TO 170
      DI=DD
      TO=L-LD
      T=TO+DI
      P1=Q
560  IF(ABS(DI).LT.END)GO TO 570
      CALL THRE5(A,T,YJ,N,M,P0,KODE)
      DI=DI*0.5
      IF(P0.EQ.0)GO TO 565
      TO=T
      P1=P0
      T=T-DI
      GO TO 560
565  T=T+DI
      GO TO 560
570  CONTINUE
      T=TO
      IF(P1*P)170,170,580
580  CONTINUE
590  CONTINUE
      L11=L-LD
      ZP=A(L11,J)+(T-L+LD)*(A(L,J)-A(L11,J))/LD
      CALL THRE4(T,YJ,ZP,XP,HH,VV,KODE)
      HH=((XP-QX)*SM-(HH-QY)*SL)*SD
      VV=(VV-QZ)*SD

```

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

      HH=(HH-H(9))*SH
      VV=(VV-V(9))*SV
      IF(PEN.EQ.DOWN)CALL DRAW(HH,VV)
      IF(PEN.EQ.UP)CALL MOVE(HH,VV)
C
600  PEN=INTS(5)-PEN
      GO TO 170
C
610  CONTINUE
      PEN=DOWN
      DI=DD
      TO=L-LD
      T=TO+DI
      P1=Q
620  IF(ABS(DI).LT.END)GO TO 630
      CALL THRE5(A,T,YJ,N,M,PO,KODE)
      DI=DI*0.5
      IF(PO.EQ.0)GO TO 625
      TO=T
      P1=PO
      T=T+DI
      GO TO 620
C
625  T=T-DI
      GO TO 620
C
630  CONTINUE
      T=TO
      IF(P1*Q)600,600,590
      CALL THRE4(XI,YJ,A(L,J),XP,HH,VV,KODE)
      VV=(VV-QZ)*SD
      HH=((XP-QX)*SM-(HH-QY)*SL)*SD
190  HH=(HH-H(9))*SH
200  VV=(VV-V(9))*SV
      IF(PEN.EQ.DOWN)CALL DRAW(HH,VV)
      IF(PEN.EQ.UP)CALL MOVE(HH,VV)
1020 Q=P
1030 CONTINUE
C
      L=L+LD
      LD=-LD
      DD=-DD
1060 CONTINUE
C
1090 IF(K-3)2060,1110,2060
C
      DRAW LINES ALONG X AXIS...
C
1110 CONTINUE
C
      L=0
      LD=1
      DD=0.5*LD
1140 DO 2040 I=1,N
      XI=I
      Q=0
1160 DO 2020 J=1,MM
      L=L+LD
      YJ=L

```

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

      CALL THRE5(A,XI,YJ,N,M,P,KODE)
      PEN=UP
      IF(P)1510,1520,1530
1510  IF(Q)1540,1550,1540
1520  IF(Q)1610,2010,1610
1530  IF(Q)1540,1550,1540
1540  CONTINUE
C
      PEN=DOWN
      GO TO 1170
1550  CONTINUE
      IF(J.EQ.1)GO TO 1170
      DI=DD
      TO=L-LD
      T=TO+DI
      P1=Q
1560  IF(ABS(DI).LT.END)GO TO 1570
      CALL THRE5(A,XI,T,N,M,PO,KODE)
      DI=DI*0.5
      IF(PO.EQ.0)GO TO 1565
      TO = T
      P1=PO
      T=T-DI
      GO TO 1560
C
1565  T=T+DI
      GO TO 1560
C
1570  CONTINUE
      T=TO
      IF(P1*P)1170,1170,1580
1530  CONTINUE
1590  CONTINUE
      L22=L-LD
      ZP=A(I,L22)+ (T-L+LD)*(A(I,L) - A(I,L22))/LD
      CALL THRE4(XI,T,ZP,XP,HH,VV,KODE)
      HH=((XP-QX)*SM - (HH - QY)*SL)*SD
      VV=(VV-QZ)*SD
      HH=(HH-H(9))*SH
      VV=(VV-V(9))*SV
      IF(PEN.EQ.DOWN)CALL DRAW(HH,VV)
      IF(PEN.EQ.UP)CALL MOVE(HH,VV)
C
1600  PEN=INTS(5)-PEN
      GO TO 1170
C
1610  CONTINUE
      PEN=DOWN
      DI=DD
      TO=L-LD
      T=TO+DI
      P1=Q
1620  IF(ABS(DI).LT.END)GO TO 1630
      CALL THRE5(A,XI,T,N,M,PO,KODE)
      DI=DI*0.5
      IF(PO.EQ.0)GO TO 1625
      TO=T
      P1=PO
      T=T+DI

```

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

      GO TO 1620
1625 T=T-DI
      GO TO 1620
1630 CONTINUE
      T=TO
      IF(P1*Q)1600,1600,1590
1170 CALL THRE4(XI,YJ,A(I,L),XP,HH,VV,KODE)
      HH=((XP-QX)*SM-(HH-QY)*SL)*SD
      VV=(VV-QZ)*SD
1180 HH=(HH-H(9))*SH
1190 VV=(VV-V(9))*SV
      IF(PEN.EQ.DOWN)CALL DRAW(HH,VV)
      IF(PEN.EQ.UP)CALL MOVE(HH,VV)
2010 Q=P
2020 CONTINUE
C
      L=L+LD
      LD=-LD
      DD=-DD
2040 CONTINUE
C
2060 CONTINUE
C
2080 RETURN
      END
C
C
C
C
C
C
      SUBROUTINE THRE5(Z,XI,YJ,M,N,P,KODE)
C
      COMMON ANGA,ANCB,HV,D,SH,SV
      COMMON SL,SM,SN,CX,CY,CZ,QX,QY,QZ,SD
      REAL*4 ANGA,ANCB,HV,D,SH,SV,SL,SM,SN,CX,CY,CZ,QX,QY,QZ,SD
      REAL*4 Z,XI,YJ
      REAL*4 ZB,XEND,DX,YMULT,ZMULT,YEND,DY,XMULT,XB,YB,XSTEP,YSTEP
      REAL*4 ZS,SGN
      INTEGER*2 M,N,KODE,IR,JC,IDX,JDY
      INTEGER CUM,CNT,P
      REAL I,J,II,JJ
      DIMENSION Z(192,192)
C
C
C
      SEE IF A POINT IS VISIBLE
C
      IF(KODE.EQ.1)GO TO 78
      IR=XI
      JC=YJ
      ZB=Z(IR,JC)
      IF(XI.EQ.IR)GO TO 2
      ZB=Z(IR,JC)+(XI-IR)*(Z(IR+1,JC)-Z(IR,JC))
      GO TO 4
C
C
C
      IF(YJ.EQ.JC)GO TO 4
      ZB=Z(IR,JC)+(YJ-JC)*(Z(IR,JC+1)-Z(IR,JC))
      CONTINUE
      XEND=0.
      DX=0.
      YMULT=0.

```


C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```

      ZMULT=0.
      IF(XI.EQ.CX)GO TO 10
      YMULT=(YJ-CY)/(XI-CX)
      ZMULT=(ZB-CZ)/(XI-CX)
      DX=1.
      XEND=M+1
C
      IF(XI.LT.CX)GO TO 10
      DX=-1.
      XEND=0.
10    CONTINUE
      YEND=0.
      DY=0.
      XMULT=0.
      IF(YJ.EQ.CY)GO TO 20
C
      XMULT=(XI-CX)/(YJ-CY)
      IF(ZMULT.EQ.0.) ZMULT=(ZB-CZ)/(YJ-CY)
      DY=1.
      YEND=N+1
      IF(YJ.LT.CY)GO TO 20
C
      DY=-1.
      YEND=0.
C
20    CONTINUE
      CUM=0
      CNT=0
      P=0
      XB=XI
      YB=YJ
C
30    CONTINUE
      II=AINT(X)
      JJ=AINT(Y)
      XSTEP=DX
      YSTEP=DY
C
      IF(XB.EQ.II)GO TO 40
      IF(DX.LT.0.) XSTEP=0.
      GO TO 45
C
40    IF(YB.EQ.JJ)GO TO 45
      IF(DY.LT.0.) YSTEP=0.
45    CONTINUE
      I=II+XSTEP
      J=JJ+YSTEP
      IF(I.EQ.XEND)GO TO 80
      IF(J.EQ.YEND)GO TO 80
C
      XB=CX+XMULT*(J-CY)
      YB=CY+YMULT*(I-CX)
C
      IF(DX.LT.0.)GO TO 55
      IF(XB.LT.I)GO TO 60
C
50    XB=I
      GO TO 65
C
```

C *** TWO DIMENSIONAL SURFACE PLOTTING PROGRAM ***

```
53 IF(XB.LT.1)GO TO 50
54 YB=J
55 CONTINUE
   ZB=CZ+ZMULT*(XB-CX.
   IR=I
   JC=J
   IF(YB.NE.J)GO TO 70
   IDX=I-DX
   ZS=Z(IR,JC)-DX*(XB-I)*(Z(IDX,JC) - Z(IR,JC))
   GO TO 75
C
70 JDY=J-DY
   ZS=Z(IR,JC)-DY*(YB-J)*(Z(IR,JDY) - Z(IR,JC))
C
75 CONTINUE
   SGN=1.
   IF(ZB.LT.ZS) SGN=-1.
   CUM=CUM+INT(SGN)
   CNT=CNT+1
   IF(IABS(CUM).EQ.CNT)GO TO 30
   GO TO 90
C
78 P=1
   GO TO 95
C
80 CONTINUE
   P=1
   IF(CUM) 84,86,90
84 P=-1
   GO TO 90
86 CONTINUE
   IF(ZB.LE.CZ)GO TO 90
   P=-1
90 CONTINUE
C
95 RETURN
   END
```

AD-A138 142

EMPLOYMENT OF ADAPTIVE LEARNING TECHNIQUES FOR THE
DISCRIMINATION OF ACQU. (U) GENERAL ELECTRIC CORPORATE
RESEARCH AND DEVELOPMENT SCHENECTA. J W ERKES ET AL.
NOV 83 83-SRD-060 N00014-82-C-2031

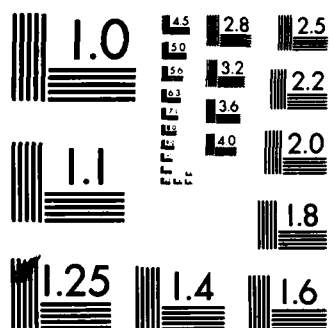
4/4

UNCLASSIFIED

F/G 20/1

NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

7.9 Example Dereverberating a Lamb Wave Convolved with a Resonant Sensor

7.9 Example Dereverberating a Lamb Wave Convolved With a Resonant Sensor

The simulation performed here to test the deresonating characteristics of homomorphic filtering involved the output of a resonant acoustic emission sensor to a seismic surface pulse. The pulse used was the Pekeris solution [83,84,85,86] of a Lamb surface pulse and the sensor impulse response was modeled as a decaying sinusoid. Figure 7.9.1 shows the Lamb wave and the sensor response. The signal received from the sensor is the convolution of these two signals which is shown in figure 7.9.2. Figure 7.9.3 shows the spectrum of the convolved response and figure 7.9.4 is the cepstrum of this same signal. Homomorphic filtering the cepstrum by setting to zero the segment of the quefrency which can be attributed mostly to the resonant sensor response (the high quefrency section) and inverting the resulting cepstrum results in figure 7.9.5 which as expected resembles the Lamb wave. Conversely filtering out the Lamb wave using this same method leaves the resonant sensor response as shown in figure 7.9.6. These individual responses could be optimized by adjusting the cut-off quefrency in the homomorphic filtering step. Many possibilities exist for optimized signal energy level cut off points and tapered attenuation near cut off points rather than hard attenuation.

The nonideal reconstruction of the pulse and ringing sensor signals is due to the fact that there is no perfect separation in the cepstrum of the pulse and decaying oscillations as there had

been between the previously studied pulses and the channels with discrete echoes. The channel is modeled as

$$h(t) = e^{-t/\tau} \cos(\omega_c t) \quad (7.9.1)$$

whose Fourier transform is

$$H(\omega) = \frac{j\omega\tau^2}{1 + 2j\omega\tau + (\omega_c^2 - \omega^2)\tau^2} \quad (7.9.2)$$

Taking the log of eq. (7.9.2) yields

$$\log [H(\omega)] = \log (j\omega\tau^2) - \log (1 + 2j\omega\tau + (\omega_c^2 - \omega^2)\tau^2) \quad (7.9.3)$$

$$= \log (j\tau^2) + \log (\omega) - \log (1 + 2j\omega\tau + (\omega_c^2 - \omega^2)\tau^2) \quad (7.9.4)$$

and the cepstrum equals

$$H(\tau) = \log (j\tau^2) \delta(\tau) + F^{-1}[\log(\omega)] + F^{-1}[\log (1 + 2j\omega\tau + (\omega_c^2 - \omega^2)\tau^2)] \quad (7.9.5)$$

The term $F^{-1}[\log(\omega)]$ in (7.9.5) prevents the ideal separation between the channel and input pulse in this case due to its significant "DC" component which appears at $\tau = 0$ in the cepstrum, however as seen by the results, this overlapping does not appear to be very severe. The approximate form of the Lamb wave used in this simulation was a logarithm pulse

$$p(t) = \begin{cases} -\log_{10} (1 + \epsilon - t) & 0 < t < 1 \\ -1 & \text{elsewhere} \end{cases} \quad (7.9.6)$$

which is difficult to analyze and could easily contain higher frequency components that overlap the channel response and also prevent separation. This is very likely since the pulse shape maintains a constant value over a finite period of time and therefore would contain substantial high frequency components. Calculating the transform of the log component by

$$P_{\ell}(\omega) = \int_{\epsilon}^{1+\epsilon} -\log_{10}(-t) e^{-j\omega t} dt \quad (7.9.7)$$

$$\approx \int_{\epsilon}^1 -2.3 \log(-t) e^{-j\omega t} dt \quad (7.9.8)$$

$$\approx -2.3 \int_{\epsilon}^1 \left(-t - \frac{t^2}{2} - \frac{t^3}{3} - \dots\right) e^{-j\omega t} dt \quad (7.9.9)$$

$$\approx 2.3 \int_{\epsilon}^1 t e^{-j\omega t} dt + \dots \quad (7.9.10)$$

and evaluating the first definite integral term yields

$$P_{\ell}(\omega) = 2.3(j\omega)^2 [e^{-j\omega} (j\omega - 1) - e^{-j\omega\epsilon} (j\omega\epsilon - 1)] + \dots \quad (7.9.11)$$

Taking the log of $P_{\ell}(\omega)$ results in a $\log(\omega)$ term which also appeared in the log of $H(\omega)$ for the channel. This demonstrates at least one area of overlap between the cepstra of the pulse and the channel.

Figure 7.9.4 can be divided into 3 separate regions for further analysis. The first is the pulse from the origin to $\tau = 100$

which contains the log pulse shape and some discrete effects due to the DFT. The second region is that around the negative pulse at $\tau \approx 150$. This is due to the resonant frequency of the transducer. The oscillations in the high quefreny segment form the third region and are due to a combination of the pulse shape, the decaying resonant transducer response and discrete effects.

Overall, the results are very promising for real and practical dereverberation applications where it is desired to obtain an input pulse shape from a narrow band or resonant transducer. More work is needed in this area to determine optimized filter characteristics and the effect of noise on this type of signal processing.

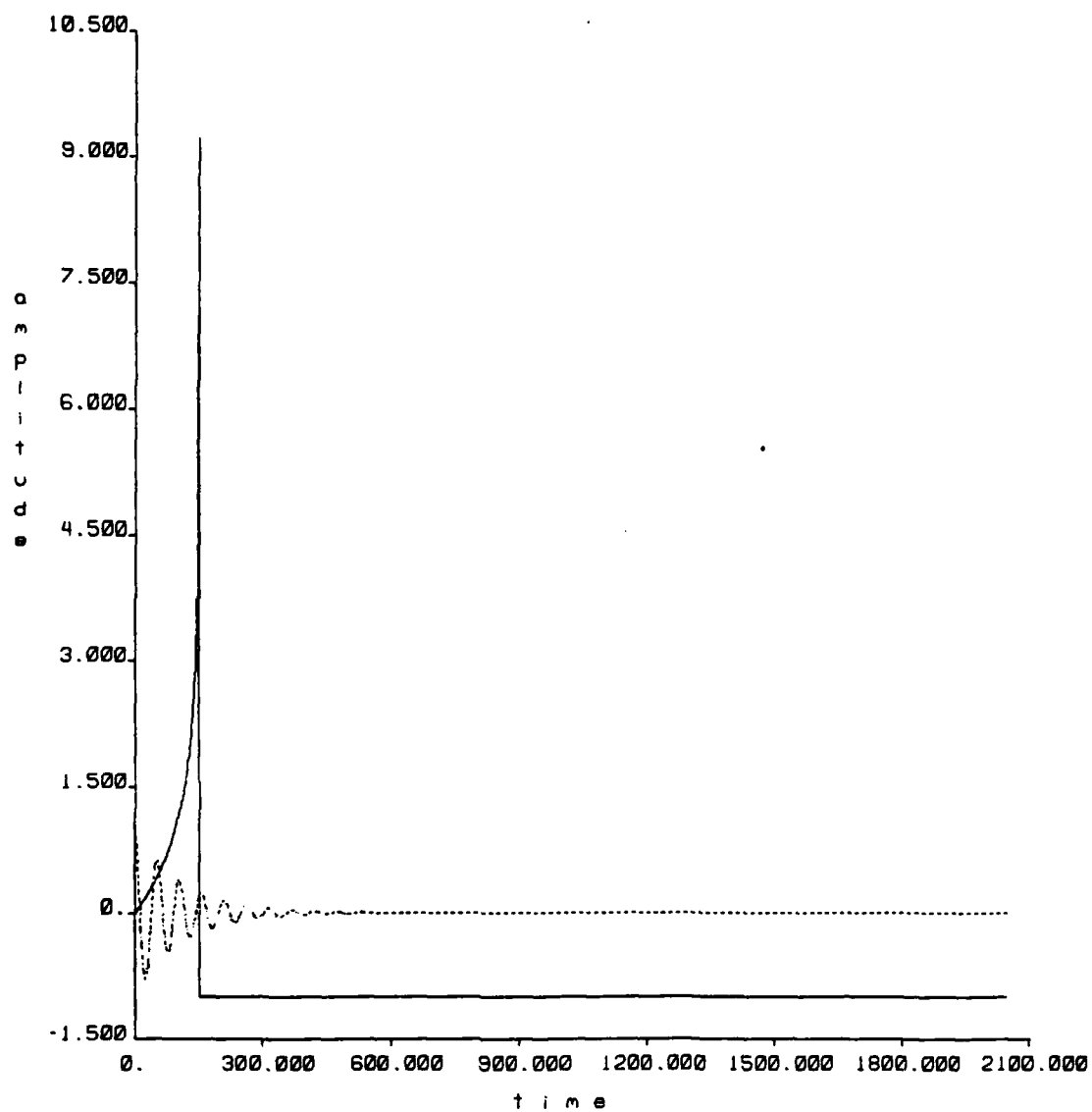


Figure 7.9.1 Lamb Wave (solid line) and Resonant Sensor Impulse Response (dashed line)

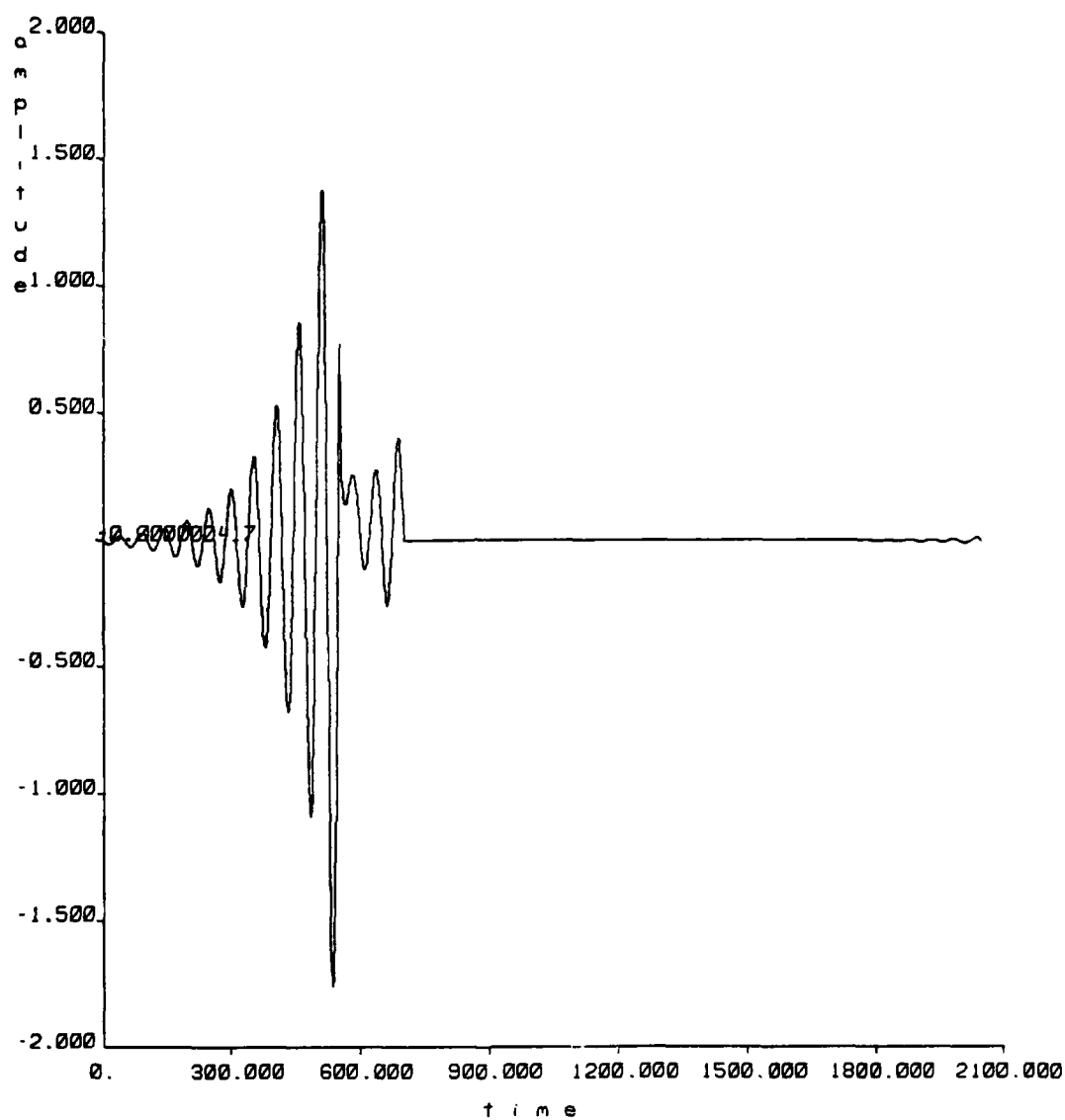


Figure 7.9.2 Convolved Lamb Wave and Sensor Response

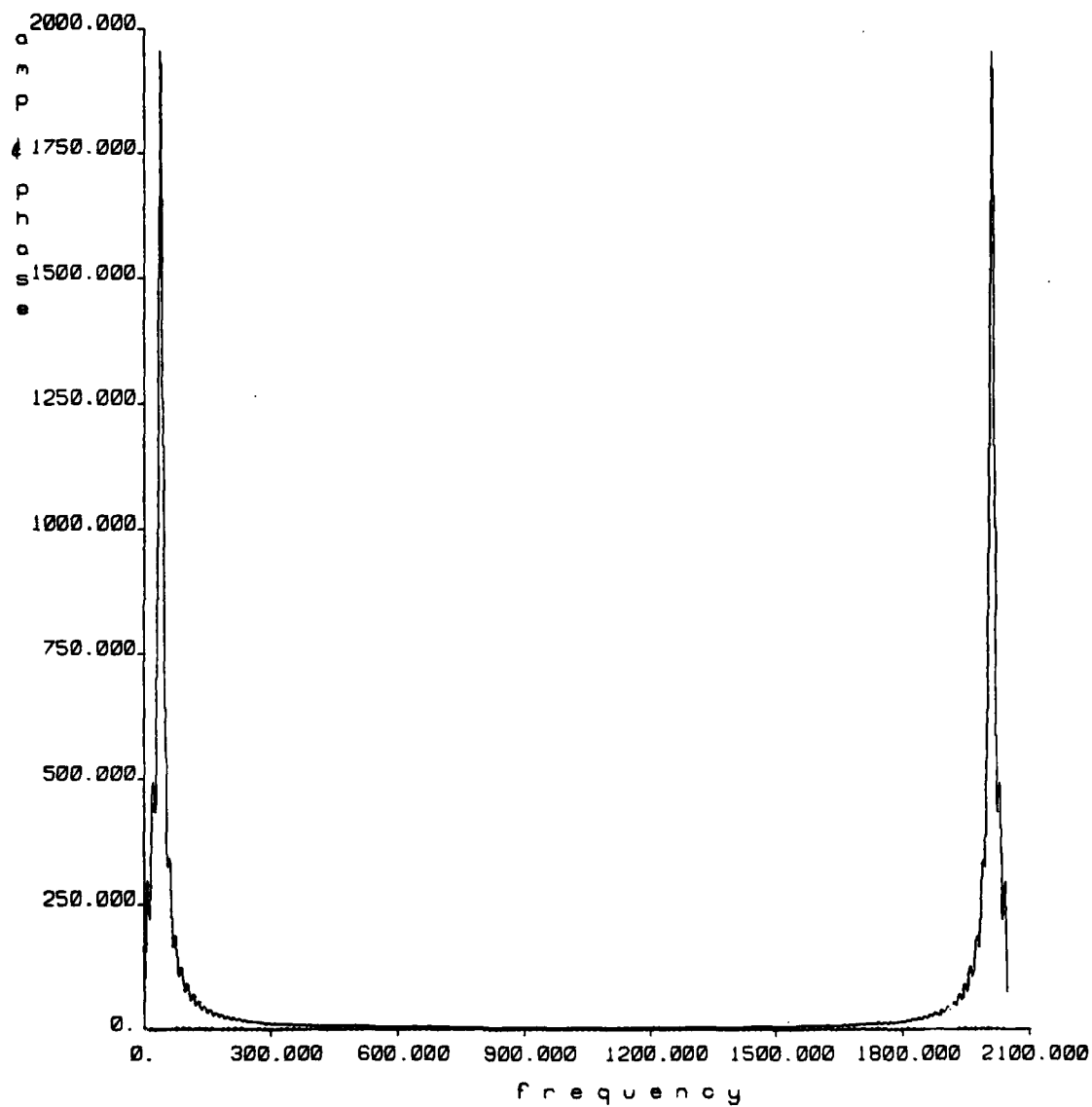


Figure 7.9.3 Fourier Transform of Convolved Lamb Wave and Sensor Response

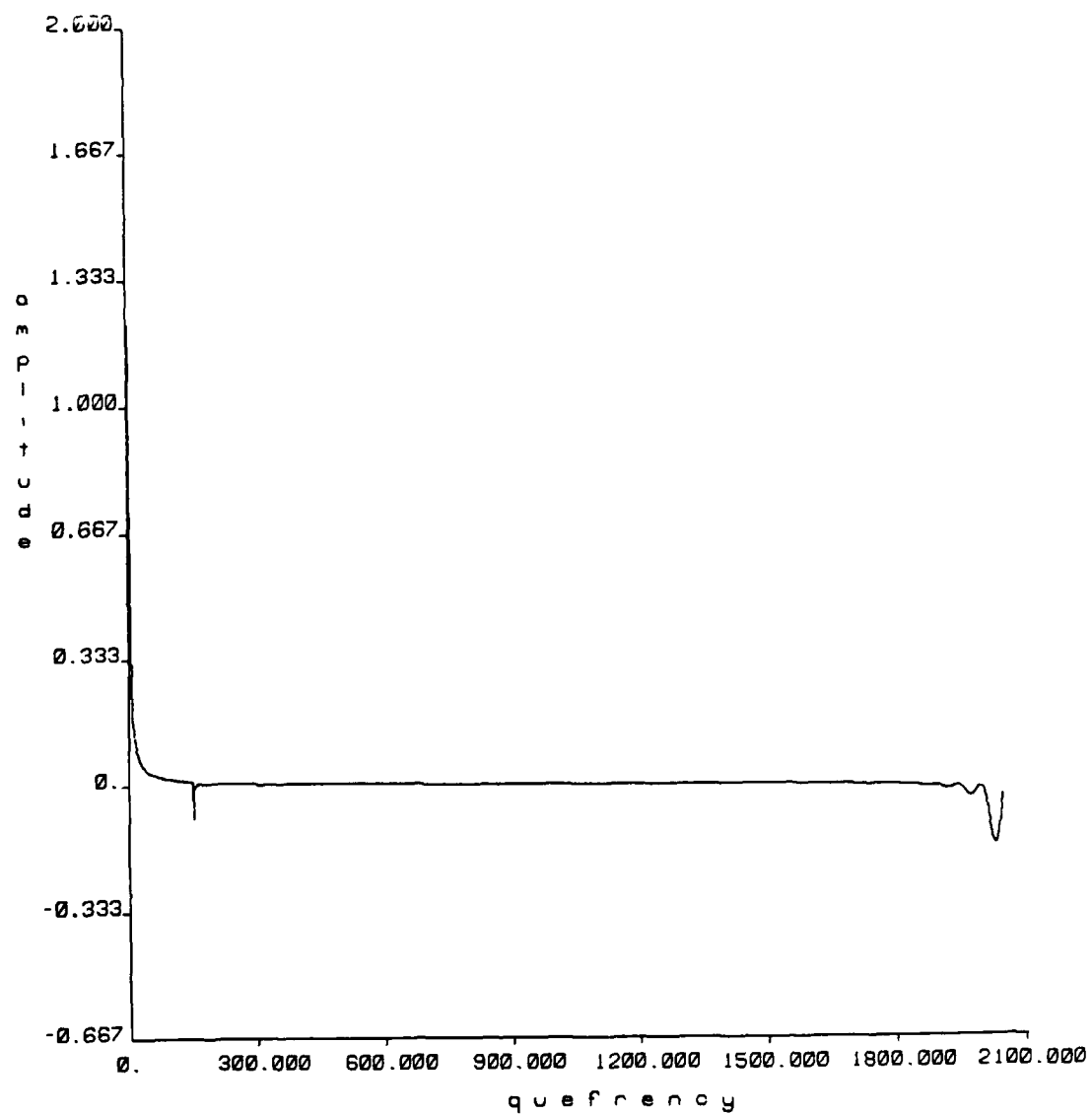


Figure 7.9.4 Cepstrum of Convolved Lamb Wave and Sensor Response

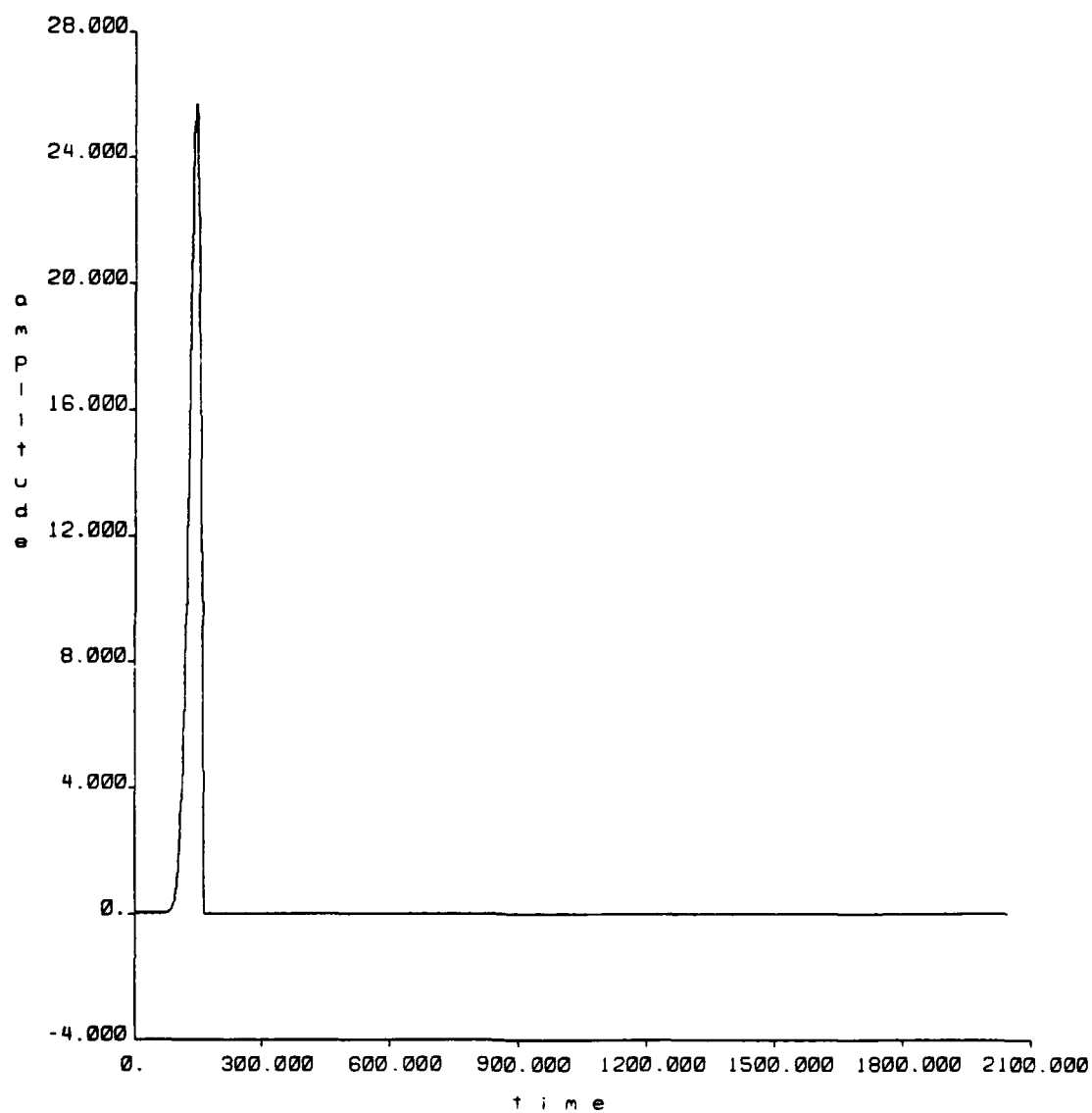


Figure 7.9.5 Dereverberated Lamb Wave

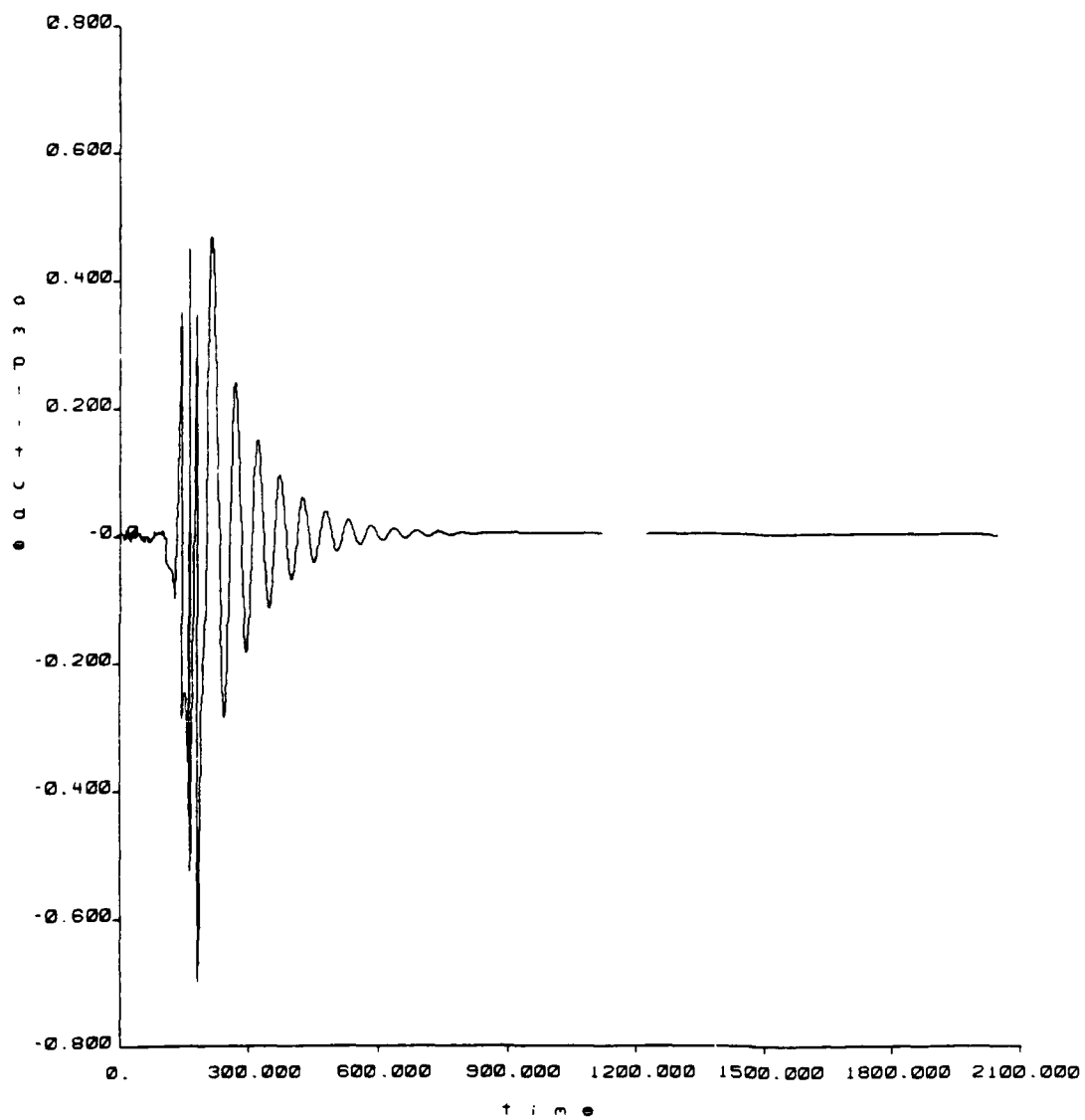


Figure 7.9.6 Sensor Time Signal Remaining After Filtering Out Lamb Wave

Appendix B

Appendix B

SIGNAL PROPAGATION PHYSICS IN A BOUNDED ELASTIC MEDIUM AND THE INTERPRETATION OF ACOUSTIC EMISSION SIGNATURES

B.1 INTRODUCTION

It is possible to make a physically meaningful interpretation of an acoustic emission (AE) signal which has been de-reverberated by means of a homomorphic deconvolution procedure employing cepstral techniques⁽¹⁾ provided that the specimen ringdown transfer function H_r is a function only of frequency ω . One may not assume *a priori* that this condition is satisfied when performing an AE experiment, however, because in a given specimen having a prescribed shape, on which the location of a transducer has been specified, the ringdown transfer function can be expected to vary with the location within the specimen of the events that give rise to the signals received by the transducer. A further complication arises from the fact that many kinds of AE events (e.g., crack propagation) are directional in nature and can be expected to give rise to directional acoustic radiation. As a consequence the ringdown transfer function can be expected to depend upon both the orientation and the location of an AE event in three-dimensional space.

While these complications pose no particular problem to an experimenter when both the location and orientation of each AE event are known, the situation is quite different when this information is not available. Under these circumstances the experimenter may seek to justify his ignorance of the location and orientation of a particular AE event on the grounds that lack of information concerning these parameters may increase the uncertainty of the experimental estimate of the amplitude of the event, but that it leaves intact (after deconvolution) the *shape* of the signature, which is sufficient to identify the *nature* of the event. His argument is a valid one if he can show that the acoustic radiation pattern produced by the event in question is *non-directional*, but if the pattern is directional he cannot justify out of hand ignorance either of the position or of the orientation of the event.

The reason he cannot do this is that in general the directivity of the resulting pattern of acoustic radiation will cause the shapes of the event signatures transmitted along different paths to be different. The resulting signal that is received at a given transducer location will therefore, following de-reverberation by means of the cepstral method, be a composite sum of the signatures radiated in all directions by the event. Since the relative weighting of each of the component signatures in the composite sum is dependent upon the unknown position and orientation of the event within the specimen, the ability of the experimenter to interpret in a physically meaningful way the information contained in the de-reverberated signal received at a given transducer location may be severely limited. In principle it is possible to overcome this difficulty by employing a large array of transducers extending over a large fraction of the specimen surface, but practical considerations make this an undesirable approach, particularly in those cases in which the object of the experiment is to identify the nature of the AE event taking place without consideration of its location or orientation.

In practice it will be necessary to develop a set of guidelines for ascertaining the minimum number of transducers (and the most favorable locations for them) that

must be employed on a given specimen if sufficient information is to be obtained to permit successful identification of AE events from the transducer signatures. In the process, attention must be paid to the physics of wave propagation, both within and along the surfaces of a specimen of given shape, in order that a suitable combination of analytical and computational techniques can be set up from which these guidelines can be obtained. With this end in mind, preliminary efforts have been undertaken at the RPI Laboratory for Noise Control Research for the purpose of examining the physics of wave propagation and the suitability of various techniques described in the literature for analyzing wave propagation in bounded and unbounded elastic media. This report summarizes these activities.

B.2 WAVE PROPAGATION IN AN ELASTIC MEDIUM

In an elastic medium the equation of motion and Hooke's law combine to produce the well-known equation

$$\mu \nabla^2 \bar{u} + (\lambda + \mu) \nabla (\nabla \cdot \bar{u}) = \rho \frac{\partial^2 \bar{u}}{\partial t^2} \quad (1)$$

where \bar{u} is the particle displacement, λ and μ are Lamé's constants, and ρ is the mass density of the medium. It is customary in analysis of wave propagation to neglect the effects of mechanical dissipation, and this neglect is reflected in the form of (1), which contains no dissipative (i.e., first-order time derivative of \bar{u}) term. Equation (1) is readily transformed into a scalar and a vector wave equation if one writes \bar{u} in terms of a scalar potential ϕ and a vector potential ψ according to the relation

$$\bar{u} = -\nabla \phi + \nabla \times \psi \quad (2)$$

Substitution of (2) into (1) yields a result that is separable into a scalar wave equation

$$c_p^2 \nabla^2 \phi = \frac{\partial^2 \phi}{\partial t^2} \quad (3)$$

and a vector wave equation

$$c_s^2 \nabla^2 \psi = \frac{\partial^2 \psi}{\partial t^2} \quad (4)$$

Here $c_p^2 = (\lambda + 2\mu)/\rho$ and $c_s^2 = \mu/\rho$ are the squares of the respective plane-wave propagation speeds (assuming the medium is unbounded) of the ϕ -wave and the ψ -wave (referred to in geophysics as the p -wave and the s -wave). The separation of (1) in this manner is well-known in the literature, where it is referred to as the Helmholtz Decomposition Theorem.

The simple picture presented by equations (3) and (4) is transformed into one of monumental complexity when boundaries are introduced, since multiple reflections will generate an elaborate pattern of vibrations even when only a single source is present. For example, the interactions of p - and s -waves with a single plane surface in an otherwise infinite elastic medium will produce Rayleigh waves⁽²⁾ which move along the surface and are nondispersive in character. If the medium is inhomogeneous or the boundary is loaded mechanically by another medium, even more complicated phenomena such as Love waves^(3,4) may be produced. Analysis of the motion of the boundary has long been a part of geophysics and seismology, but early work such as that of Lamb⁽⁵⁾ were of necessity restricted to comparatively simple cases. More recent work in this area has sought to integrate the analysis of fairly elaborate models of

the interior of the earth—see, for example, Aki and Richards⁽⁶⁾ and Ben-Menachem et al⁽⁷⁾—with sophisticated methods of signature analysis such as are described by Robinson and Treitel⁽⁸⁾ and Webster.⁽⁹⁾

In contrast to the direction taken by geophysics and seismology, the literature of acoustic emission as applied to nondestructive testing (NDT) techniques has to a great extent been limited to the practical problems posed by the response characteristics of transducers and the presence of background noise. Typical examples of the literature concerned with transducers are given in the literature,⁽¹⁰⁻¹²⁾ as are descriptions of the methodology of NDT involving acoustic emission.⁽¹³⁻¹⁵⁾ Some work involving the methodology employed in geophysics and seismology, described above, has been done,^(16,17) but it is significant that as recently as 1980 Eisenblatter⁽¹⁸⁾ observed that the current practice in NDT, insofar as AE signature analysis was concerned, took little or no cognizance of the role played by the directional acoustic radiation pattern of an event. More recent efforts have addressed the problem of directivity, e.g., the work of Ceranoglu and Pao⁽¹⁹⁾ which treats acoustic emission arising from directional-source models of acoustic emission events such as concentrated forces or couples.

The propagation of acoustic vibrations through an elastic solid has long been a subject of study, being discussed in a major work by Love⁽²⁰⁾ as far back as 1892. A more modern treatise of considerable importance is the work of Mindlin,⁽²¹⁾ which deals extensively with the vibration of plates from the standpoint of modal analysis. This approach has been used by Weaver and Pao⁽²²⁾ to assess the response of a plate to a time-varying point load. Another method of analysis, especially useful when studying the transient propagation of acoustic radiation through an elastic solid, is the so-called *generalized ray theory* (see Pao⁽²³⁾ for a recent treatment of this subject). Both of these approaches have their strong points and their weak points, and one of the tasks of the work described in this report was a cursory examination of both approaches, in light of these strengths and weaknesses, in order to assess on a comparative basis their suitability as tools for evaluating the ringdown transfer function H_r .

B.3 MODAL ANALYSIS

Of the two approaches for analyzing wave propagation mentioned in the previous section, the one which was developed earliest was that involving modal analysis. The method of Weaver and Pao⁽²²⁾ for axisymmetric waves in a uniform plate of constant thickness $2h$ was examined, and the results of that examination are contained in Appendix A. In essence, the method involves the summation of a set of normalized modes, multiplied by constants so chosen that the summation yields the axisymmetric response of a plate to a concentrated force of magnitude Q_0 applied in the z direction—i.e., perpendicular to the parallel surfaces of the plate—a distance z_0 from the midplane of the plate cross section (see Figure 1 for details of the coordinate system).

The timewise behavior of the force is taken to be that of a step function $H(t)$, and in Weaver and Pao⁽²²⁾ the particle displacement \bar{u} is shown to be given by

$$\bar{u}(r, z, t) = H(t) \sum_{m=1}^{\infty} \int_0^{\infty} f_m(z_0, \kappa) \frac{\bar{u}_m(r, z, \kappa)}{M_m(\kappa)} \frac{1 - \cos[\omega_m(\kappa)t]}{\omega_m^2(\kappa)} d\kappa \quad (5)$$

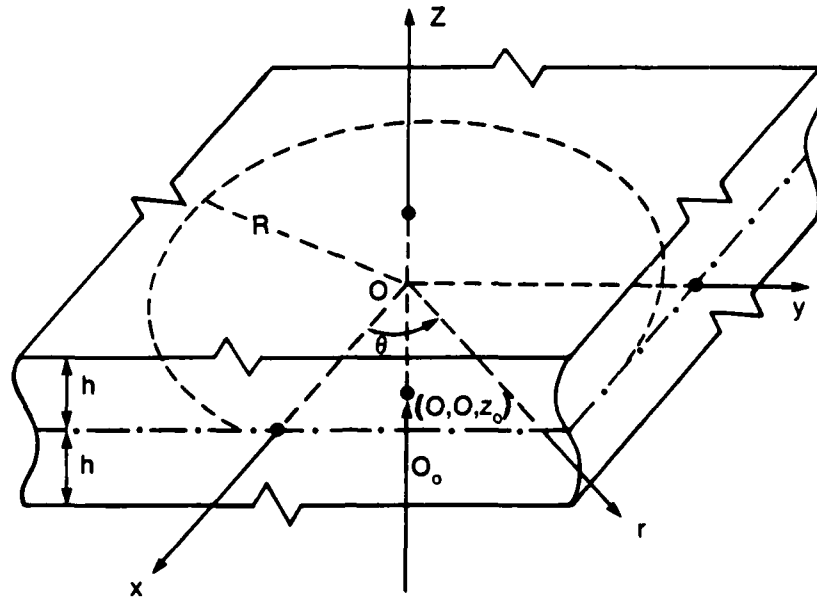


Figure 1. Axisymmetric force in a thick plate

where κ is the wave number and f_m is a modal weighting factor given by

$$f_m(z_0\kappa) = Q_0 \kappa^2 [2\alpha_m \beta_m \sin \beta_m h \cos \alpha_m z_0 + (\kappa^2 - \beta_m^2) \sin \alpha_m h \cos \beta_m z_0] \quad (6)$$

if $m=1,3,5,\dots$, and by

$$f_m(z_0\kappa) = Q_0 \kappa^2 [2\alpha_m \beta_m \cos \beta_m h \sin \alpha_m z_0 + (\kappa^2 - \beta_m^2) \cos \alpha_m h \sin \beta_m z_0] \quad (7)$$

if $m=2,4,6,\dots$, while the r -component U_m and the z -component W_m of the m th mode shape are given respectively by

$$U_m = \beta_m [2\kappa^2 \sin \beta_m h \sin \alpha_m z - (\kappa^2 - \beta_m^2) \sin \alpha_m h \sin \beta_m z] \kappa J'_0(\kappa r) \quad (8)$$

and

$$W_m = \kappa^2 [2\alpha_m \beta_m \sin \beta_m h \cos \alpha_m z + (\kappa^2 - \beta_m^2) \sin \alpha_m h \cos \beta_m z] J_0(\kappa r) \quad (9)$$

when $m=1,3,5,\dots$, and

$$U_m = \beta_m [-2\kappa^2 \cos \beta_m h \cos \alpha_m z + (\kappa^2 - \beta_m^2) \cos \alpha_m h \cos \beta_m z] \kappa Y'_0(\kappa r) \quad (10)$$

and

$$W_m = \kappa^2 [2\alpha_m \beta_m \cos \beta_m h \sin \alpha_m z + (\kappa^2 - \beta_m^2) \cos \alpha_m h \sin \beta_m z] J_0(\kappa r) \quad (11)$$

when $m=2,4,6,\dots$. Here J_0 and J'_0 denote the zeroth-order Bessel function of the first kind and its derivative, respectively, while ω_m is related to α_m and β_m by the expressions

$$\alpha_m^2 = \omega_m^2 / c_p^2 - \kappa^2 \quad (12)$$

and

$$\beta_m^2 = \omega_m^2 / c_s^2 - \kappa^2 \quad (13)$$

and, for a given value of κ , ω_m is the m th root of the transcendental dispersion equation

$$0 = (\kappa^2 - \beta^2)^2 \cos \alpha h \sin \beta h \quad (14)$$

$$+ 4\alpha\beta\kappa^2 \cos \beta h \sin \alpha h$$

with $\alpha^2 = \omega^2 / c_p^2 - \kappa^2$ and $\beta^2 = \omega^2 / c_s^2 - \kappa^2$. The quantity M_m given by

$$M_m = 2\rho \left\{ B_m^2 [\omega_m^2 h / c_p^2 + (\alpha_m^2 - \kappa^2) \sin 2\alpha_m h / 2\alpha_m] + C_m^2 \kappa^2 [\omega_m^2 h / c_s^2 - (\beta_m^2 - \kappa^2) \sin 2\beta_m h / 2\beta_m] + 4B_m C_m \kappa^2 \sin \alpha_m h \cos \beta_m h \right\} / \kappa \quad (15)$$

when $m=1,3,5,\dots$, and by

$$M_m = 2\rho \left\{ A_m^2 [\omega_m^2 h / c_p^2 - (\alpha_m^2 - \kappa^2) \sin 2\alpha_m h / 2\alpha_m] + D_m^2 \kappa^2 [\omega_m^2 h / c_s^2 + (\beta_m^2 - \kappa^2) \sin 2\beta_m h / 2\beta_m] + 4A_m D_m \kappa^2 \cos \alpha_m h \sin \beta_m h \right\} / \kappa \quad (16)$$

when $m=2,4,6,\dots$, serves to normalize the mode shapes. Here

$$A_m = -2\kappa^2 \beta_m \cos \beta_m h \quad (17)$$

$$B_m = 2\kappa^2 \beta_m \cos \beta_m h$$

$$C_m = (\kappa^2 - \beta_m^2) \sin \alpha_m h$$

$$D_m = (\kappa^2 - \beta_m^2) \cos \alpha_m h$$

Though mathematically very complex, such a modal analysis has the advantage that an explicit functional relation can be derived once the mode shapes have been established for a given set of boundary conditions, and for those cases in which the mode shapes can be established in this fashion some very real savings in computer time should be possible. Weaver and Pao⁽²²⁾ point out that by using the dispersion relation (14) to change the variable of integration from κ to ω , use can be made of fast Fourier transform (FFT) techniques to economize on computer time when evaluating the integral in (5). Computer programs, listed in Appendix A, were written to assess the feasibility of this approach by solving some of the examples given in Weaver and Pao⁽²²⁾ and comparing the computed results with those described in this reference. Because of problems caused by lack of computational accuracy, and in view of the limited budget allowance for computer time, it was not possible to do more than show qualitative agreement between the computed results and those described in Weaver and Pao.⁽²²⁾ Nevertheless it is believed that with computational refinements the method of modal analysis represents a feasible approach for handling acoustic emission wave propagation problems in those circumstances in which the mode shapes are easily obtainable.

B.4 GENERALIZED RAY METHOD

In contrast to the method of modal analysis, the generalized ray theory requires no prior knowledge of functions like mode shapes, which are peculiar to a given boundary geometry, and can be used with the same facility whether the source is directional or nondirectional. Its application, hereafter referred to as the *generalized ray method*, requires that one know all the paths along which a disturbance may travel and still arrive (within some finite time) at the point where a transducer is located. This information is obtained by constructing paths, or "rays" that start out from the source and then interact repeatedly with the boundaries of the elastic medium, with the interactions giving rise to reflections back into the medium and transmissions into a second medium, if one is in contact with the boundary.

Each time a ray intersects with a boundary, two rays will be reflected back into the medium and two more will be transmitted into the second medium, if one is present. The reason for the doubling of rays is that each incident ray, representing the path of either a p -wave or an s -wave, will produce both a p -wave and an s -wave that are reflected back into the medium as well as a similar pair of waves that are transmitted if a second medium is in contact with the boundary. In this work it is assumed for the sake of simplicity that no second medium exists, so that only reflected waves are present. Also, for the same reason and to permit future comparison with the modal analysis method described in the previous section, only axisymmetric wave patterns in a uniform plate due to a concentrated force applied to the upper plate surface are considered.

Under the above circumstances the r - and z -components u_r and u_z , respectively, of the particle displacement \bar{u} are given by

$$u_r = \frac{\partial \phi}{\partial r} - \frac{\partial^2 \psi}{\partial r \partial z} \quad (18)$$

$$u_z = \frac{\partial \phi}{\partial z} + \frac{\partial^2 \psi}{\partial z^2} - \frac{1}{c_s^2} \frac{\partial^2 \psi}{\partial r^2}$$

where ϕ and c_s are as described in the beginning of Section B.2. Only the θ -component of the vector potential ψ is nonzero, and this component has been set equal to minus the partial derivative of the scalar ψ appearing in the above equations. Both ϕ and ψ will be functions of r , z , and t , and both will satisfy scalar wave equations—equation (3) for ϕ and

$$c_s^2 \nabla^2 \psi = \frac{\partial^2 \psi}{\partial t^2} \quad (19)$$

for ψ . In both equations, ∇ denotes the differential operation $\partial^2/\partial r^2 + (1/r)\partial/\partial r$.

The wave equations can be changed from partial differential to linear algebraic forms by means of a Laplace and a Hankel transform. If the inverse Hankel transform is then applied to the solutions of these linear equations, the results are as follows:

$$\Phi(r, z, s) = s^2 \int_0^\infty \bar{F}(s) S_p(\xi) e^{-s\eta|z-z_0|} J_0(s\xi r) \xi d\xi \quad (20)$$

and

$$\Psi(r, z, s) = s^2 \int_0^\infty \frac{\bar{F}(s)}{-s\xi} S_v(\xi) e^{-s\zeta|z-z_0|} J_0(s\xi r) \xi d\xi \quad (21)$$

Here Φ and Ψ are the Laplace transforms of ϕ and ψ , respectively, ξr is the Hankel transform variable analogous to the Laplace transform variable s , and ζ and η are defined by the relations

$$\zeta = (\xi^2 + c_s^{-2})^{1/2} \quad (22)$$

$$\eta = (\xi^2 + c_p^{-2})^{1/2}$$

For a concentrated force $f(t)$ normal to the plate surface at $z=0$ (see Figure 2) the functions \bar{F} , S_p , and S_v are (cf. reference 23) as follows:

$$\bar{F}(s) = \frac{F(s)}{2\pi\mu s^2} \quad (23)$$

$$S_p(\xi) = \frac{\xi^2 + \zeta^2}{\Delta_r(\xi)}$$

$$S_v(\xi) = \frac{2\xi\eta}{\Delta_r(\xi)}$$

where $F(s)$ is the Laplace transform of $f(t)$ and

$$\Delta_r(\xi) = 4\eta\zeta\xi^2 - (\xi^2 + \zeta^2)^2$$

Equations (20) and (21) apply in the form shown only if the chosen path between source and transducer does not intersect any boundary. If the path does encounter a boundary, a pair of reflected waves will be generated whose magnitude and phase are

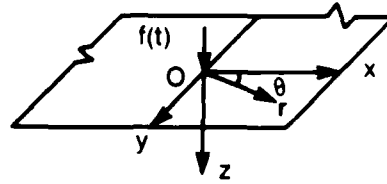


Figure 2. Line and point sources

related to the incident wave by reflection coefficients R^{pp} , R^{ss} , R^{ps} , and R^{sp} , the expressions for which are shown in reference 23 to be

$$R^{pp} = [4\eta\zeta\xi^2 + (\xi^2 + \zeta^2)^2] / \Delta, \quad (24)$$

if both the incident and the reflected wave are p -waves,

$$R^{ss} = [4\eta\zeta\xi^2 + (\xi^2 + \zeta^2)^2] / \Delta, \quad (25)$$

if both the incident and the reflected waves are s -waves,

$$R^{ps} = \pm 4\xi\eta (\xi^2 + \zeta^2) / \Delta, \quad (26)$$

if the incident wave is a p -wave and the reflected wave is an s -wave, and

$$R^{sp} = \pm 4\xi\zeta (\xi^2 + \zeta^2) / \Delta, \quad (27)$$

if the incident wave is an s -wave and the reflected wave is a p -wave. Note that when the incident and reflected waves are of the same type the reflection coefficients R^{pp} and R^{ss} are equal to each other. The positive sign is used in equations (26) and (27) when the wave propagates in the positive z direction, and the negative sign is used when the wave propagates in the negative z direction.

The first step in the application of the generalized ray method to the propagation in a uniform plate of axisymmetric waves arising from a point source consists of the construction of a set of zig-zag paths in the rz plane, each of which starts at the source location and passes through the position of the transducer. A change of direction occurs every time the path encounters the top or the bottom surface of the plate, with the angle ϵ_i of the incident path segment being related to the angle ϵ_r of the reflected path segment (the angles being measured, in each case, between the segment in question and the line normal to the surface of the plate and directed toward its interior) through Snell's law. Thus, if the incident and the reflected wave are both of the same type $\epsilon_i = \epsilon_r$, whereas if the incident wave is an s -wave and the reflected wave is a p -wave

$$\epsilon_r = \sin^{-1} [(c_s/c_p) \sin \epsilon_i]$$

while if the incident wave is a p -wave and the reflected wave is an s -wave

$$\epsilon_r = \sin^{-1} [(c_p/c_s) \sin \epsilon_i]$$

Note, however, that if ϵ_i is greater than $\sin^{-1} (c_s/c_p)$ it is impossible for an incident p -wave to give rise to a reflected s -wave.

The process of establishing each zig-zag path is one of trial-and-error that begins with an arbitrary selection of a wave type (p - or s -wave) at an arbitrary path orientation in the rz plane and then proceeds to follow this path. If the path is so orientated as to pass directly through the location of the transducer, the process is completed when the path has been followed to the transducer. If the path is not so oriented, one follows it until it reaches the plate surface. There one establishes, in accordance with the conditions outlined in the preceding paragraph, the path directions of each of the reflected waves that would arise in response to the arrival at the plate surface of the wave being followed, and then follows each of these reflected paths until it encounters either the transducer or the other plate boundary. If the other boundary is encountered the above procedure is repeated.

Thus a single original path branches into two after the first reflection, four after the second, etc., and each of these must be checked to see if it encounters the transducer location. After all paths within this set (if any) that pass through the transducer location have been identified, the entire operation is repeated, either with a different type of wave starting from the source along the same path as before or with the same type of wave traveling along another path with a different orientation. To simplify calculations, those paths that can only reach the transducer location after a very large number of reflections are usually discarded, since their contribution is generally small compared to that from a directly radiated signal or one subject to only a few reflections.

In the case under discussion here, namely axisymmetric wave propagation in a uniform plate with the source located at $r=0$ on the upper surface, paths containing small numbers of reflections are associated with initial waves that leave the source at shallow angles with respect to the plate surfaces. As the angle with respect to the plate surfaces of the initial wave is increased, the number of reflections required for the path to encounter the transducer location will increase. Because, with the exception of the case where an incident p -wave gives rise only to a reflected p -wave, each reflected path segment carries less energy than did the incident path element that gave rise to it (the remaining energy being transferred to the other reflected wave) the amplitude of the disturbance that is transmitted to the transducer location along a given path will decrease as the number of reflections in the path increases. In addition—because the path length increases with the number of reflections—as the angle between the initial wave and the plate surfaces is increased, the time required for a disturbance to be transmitted along the given path from the source to the transducer will increase.

B.5 RAY INTEGRAL EVALUATION

Once all of the zig-zag paths from the source to the receiver have been determined the Laplace transformed potentials Φ and Ψ can be inverted. The expressions for these given in equations (20) and (21) apply, as has already been observed, only for a reflection-free path, but this restriction poses no problems when reflections are present. The equations will still hold for each segment of a given path, but in every segment (except the first one) the magnitude and phase of the integrand will be affected by the reflections and distances traversed in all of the preceding segments of that path. Also, as one goes from one path segment to the next the type of wave being followed can change and, in particular, the type of wave that arrives at the transducer location *via* the final path segment may not be the same as that which started out from the source along the initial segment. As a result, when the path has been followed from source to receiver the equations will assume the following forms at the receiver location:

$$\Phi(r, Z, s) = s^2 \int_0^{\infty} \bar{K}_p(\xi) \bar{F}(s) S_p(\xi) e^{-s\eta Z} J_0(s\xi r) \xi d\xi \quad (28)$$

if the final path segment is a p -wave, and

$$\Psi(r, Z, s) = s^2 \int_0^{\infty} \bar{K}_s(\xi) \frac{\bar{F}(s)}{-s\xi} S_v(\xi) e^{-s\eta Z} J_0(s\xi r) \xi d\xi \quad (29)$$

if the final path segment is an s -wave. Here \bar{K}_p is the product of the reflection coefficients at all the points where the path whose final segment is a p -wave encountered plate boundaries, \bar{K}_s is the equivalent quantity for the path whose final segment is an s -wave, and Z is the sum of the absolute magnitudes of the z -components of the displacement of all of the path segments. (For the case being discussed, in which the source lies on the upper plate surface, Z will always be an integral number of plate thicknesses.) Because they incorporate the effects of all the reflections encountered in following the path, or "ray" from source to transducer, the integrals in (28) and (29) and all subsequent transformations of them will be referred to as *ray integrals*.

As a first step in the inversion process for Φ and Ψ , an integral form for the Bessel function J_0 ,

$$\begin{aligned} J_0(s\xi r) &= \frac{2}{\pi} \int_0^{\pi/2} \cos[(s\xi r) \cos \omega] d\omega \\ &= \frac{1}{\pi} \int_0^{\pi/2} e^{[(s\xi r) \cos \omega]} d\omega \\ &\quad + \frac{1}{\pi} \int_0^{\pi/2} e^{-[(s\xi r) \cos \omega]} d\omega \end{aligned} \quad (30)$$

is substituted into (28) and (29). As a result these equations are transformed, after some rearrangement, into

$$\Phi(r, Z, s) = \frac{s^2}{\pi} \bar{F}(s) \int_0^{\pi/2} [I_1(r, Z, s, \omega) + I_2(r, Z, s, \omega)] d\omega \quad (31)$$

and

$$\Psi(r, Z, s) = \frac{-s}{\pi} \bar{F}(s) \int_0^{\pi/2} [I_3(r, Z, s, \omega) + I_4(r, Z, s, \omega)] d\omega \quad (32)$$

where

$$\begin{aligned} I_1(r, Z, s, \omega) &= \int_0^{\infty} \bar{K}_p(\xi) S_p(\xi) e^{-s(\eta Z - \xi r \cos \omega)} d\xi \\ I_2(r, z, s, \omega) &= \int_0^{\infty} \bar{K}_p(\xi) S_p(\xi) e^{-s(\eta Z + \xi r \cos \omega)} d\xi \end{aligned} \quad (33)$$

$$\bar{I}_3(r, Z, s, \omega) = \int_0^\infty \frac{\bar{K}_s(\xi)}{\xi} S_v(\xi) e^{-s(\zeta Z - \xi r \cos \omega)} d\xi$$

$$\bar{I}_4(r, Z, s, \omega) = \int_0^\infty \frac{\bar{K}_s(\xi)}{\xi} S_v(\xi) e^{-s(\zeta Z + \xi r \cos \omega)} d\xi$$

The substitution of (33) into (31) and (32) gives the latter equations a form that is advantageous when obtaining the inverse Laplace transform of Φ and Ψ . This is because by changing the variable of integration in (33) the inverse Laplace transforms of $\bar{I}_1 - \bar{I}_4$ can be made obtainable by inspection, and once this is accomplished the inverse Laplace transforms of Φ and Ψ are easily determined by means of the convolution theorem.

The inversion process outlined above, generally known as Cagniard's method,⁽²⁴⁾ begins by setting the time t equal to the coefficient of s in each integral and making t the variable of integration. The result is

$$\bar{I}_1(r, Z, s, \omega) = \int_{Z/c_p}^\infty \bar{K}_p(\xi_1) S_p(\xi_1) e^{-st} \frac{d\xi_1}{dt} dt \quad (34)$$

$$\bar{I}_2(r, z, s, \omega) = \int_{Z/c_p}^\infty \bar{K}_p(\xi_2) S_p(\xi_2) e^{-st} \frac{d\xi_2}{dt} dt$$

$$\bar{I}_3(r, Z, s, \omega) = \int_{Z/c_s}^\infty \frac{\bar{K}_s(\xi_3)}{\xi_3} S_v(\xi_3) e^{-st} \frac{d\xi_3}{dt} dt$$

$$\bar{I}_4(r, Z, s, \omega) = \int_{Z/c_s}^\infty \frac{\bar{K}_s(\xi_4)}{\xi_4} S_v(\xi_4) e^{-st} \frac{d\xi_4}{dt} dt$$

with

$$\xi_1 = \{ \pm [(tr \cos \omega)^2 - (r^2 \cos^2 \omega - Z^2)(r^2 - Zc_p^{-2})]^{1/2} - tr \cos \omega \} / (r^2 \cos^2 \omega - z^2) \quad (35)$$

$$\xi_2 = \{ \pm [(tr \cos \omega)^2 - (r^2 \cos^2 \omega - Z^2)(r^2 - Zc_p^{-2})]^{1/2} + tr \cos \omega \} / (r^2 \cos^2 \omega - z^2)$$

$$\xi_3 = \{ \pm [(tr \cos \omega)^2 - (r^2 \cos^2 \omega - Z^2)(r^2 - Zc_s^{-2})]^{1/2} - tr \cos \omega \} / (r^2 \cos^2 \omega - z^2)$$

$$\xi_4 = \{ \pm [(tr \cos \omega)^2 - (r^2 \cos^2 \omega - Z^2)(r^2 - Zc_s^{-2})]^{1/2} + tr \cos \omega \} / (r^2 \cos^2 \omega - z^2)$$

and

$$\begin{aligned}
 \frac{d\xi_1}{dt} &= \left[\frac{Z\xi_1}{(\xi_1^2 + c_p^2)^{1/2}} - r \cos \omega \right]^{-1} \\
 \frac{d\xi_2}{dt} &= \left[\frac{Z\xi_1}{(\xi_1^2 + c_p^2)^{1/2}} + r \cos \omega \right]^{-1} \\
 \frac{d\xi_3}{dt} &= \left[\frac{Z\xi_1}{(\xi_1^2 + c_s^2)^{1/2}} - r \cos \omega \right]^{-1} \\
 \frac{d\xi_4}{dt} &= \left[\frac{Z\xi_1}{(\xi_1^2 + c_s^2)^{1/2}} + r \cos \omega \right]^{-1}
 \end{aligned} \tag{36}$$

With these substitutions the inverse Laplace transforms of $\bar{I}_1 - \bar{I}_4$ are found by inspection to be given by

$$\begin{aligned}
 I_1(r, Z, t, \omega) &= \bar{K}_p(\xi_1) S_p(\xi_1) \frac{d\xi_1}{dt} u[t - (Z/c_p)] \\
 I_2(r, Z, t, \omega) &= \bar{K}_p(\xi_2) S_p(\xi_2) \frac{d\xi_2}{dt} u[t - (Z/c_p)] \\
 I_3(r, Z, t, \omega) &= \bar{K}_p(\xi_3) S_p(\xi_3) \frac{d\xi_3}{dt} u[t - (Z/c_s)] \\
 I_4(r, Z, t, \omega) &= \bar{K}_p(\xi_4) S_p(\xi_4) \frac{d\xi_4}{dt} u[t - (Z/c_s)]
 \end{aligned} \tag{37}$$

Now that the inverse Laplace transforms are known the integrals in (31) and (32) can be evaluated numerically. The inverse Laplace transforms of the quantities outside the integral signs are $(1/\pi)d^2f/dt$ in the case of (31) and $-(1/\pi)d^2f/dt$ in the case of (32), where $f(t)$ is the time-dependent value of the applied force. The potentials Φ and Ψ can then be found by convolving these quantities with the corresponding integrals.

B.6 CONCLUSIONS

The preceding sections summarize two methods for mathematically modeling the physics of wave propagation as applied to the case of axisymmetric acoustic disturbances in an elastic plate. The generalized ray method has the advantage of greater flexibility, and it was intended that computer programs would be written to implement this method with the initial testing of the programs being carried out to simulate the response to a Pentel lead-break impulse such as that described in reference 1. The simplest case considered would be the plate of infinite thickness, the response of which was derived by Pekeris⁽²⁵⁾ and is shown in Figure 3, since this would permit comparison of the numerical results obtained from the programs. Further testing could then be performed by comparing the calculations for a plate of finite thickness with those obtained by the method of Weaver and Pao⁽²²⁾ described in Section B.3,

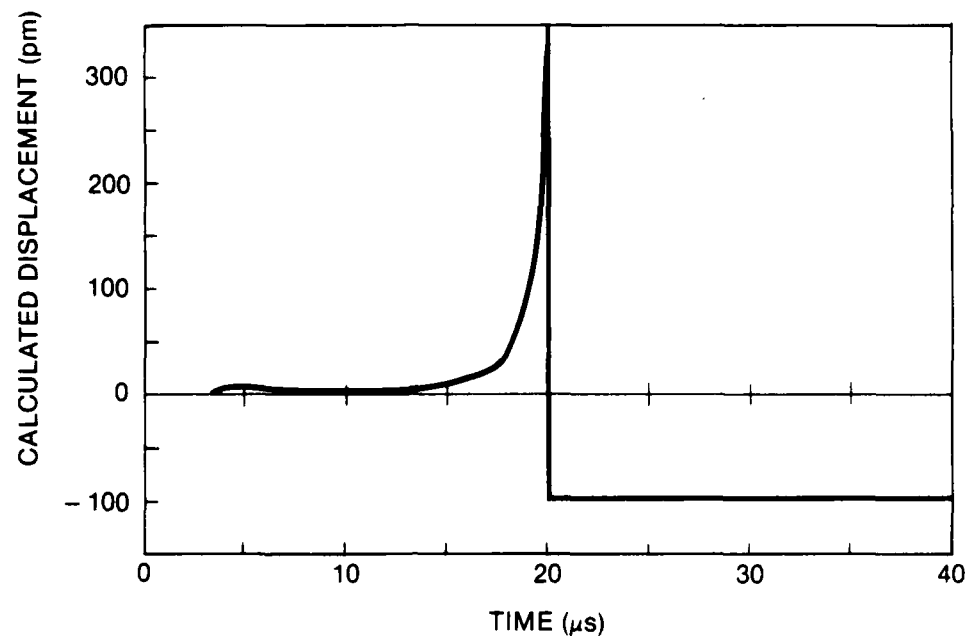


Figure 3. The seismic surface pulse according to Perkeris;⁽²⁵⁾
input, point-force-step function

after which the task of modifying the programs to include unsymmetrical sources would begin.

The problems that exist with respect to implementation of the modal analysis method have already been discussed and need not be repeated here. Suffice it to say that the primary one is the cost of the computer time necessary to generate data. It was surmised that the problem would be faced in implementing the generalized ray method, and personal communication with Y.S. Pao disclosed that over distances greater than a few plate thicknesses the number of ray integrals that would have to be evaluated would be unmanageable with any but the most powerful of the computers currently in operation. One possible avenue of escape from this dilemma would be to use the generalized ray method in the region near the source where the number of ray integrals to be evaluated is not excessive, and then let the solution obtained in this manner act as the source in a modal analysis algorithm.

B.7 REFERENCES

1. *Employment of Adaptive Learning Techniques for the Discrimination of Acoustic Emissions*, General Electric Company Technical Proposal in Response to RFP Number N00014-81-R-MA44, 1981.
2. Rayleigh, J.W.S. 3rd Baron of, "On Waves Propagated Along the Surface of an Elastic Solid," *Proceedings of the London Mathematical Society*, vol. 17, pp. 4-11, 1885.
3. Love, A.E.H., *Some Problems of Geodynamics*, New York, Dover Publications, 1967.

4. Andrianova, Z.S., Keilis-Borok, V.I., Levshin, A.L., and Neigauz, M.G., *Seismic Love Waves*, New York, Consultants Bureau (Plenum Publishing Company) 1967.
5. Lamb, Horace, "On the Propagation of Tremors over the Surface of an Elastic Solid," *Phil. Trans. Roy. Soc.*, vol. 203A, 1903, pp. 1-41.
6. Aki, Keiiti, and Richards, Paul G., *Quantitative Seismology: Theory and Methods* (2 vols.), San Francisco, W. H. Freeman and Company, 1980.
7. Ben-Menachem, Ari, and Singh, Sarva Jit, *Seismic Waves and Sources*, New York, Springer-Verlag, 1981.
8. Robinson, Enders A., and Treitel, Sven, *Geophysical Signal Analysis*, Englewood Cliffs, Prentice-Hall Inc., 1980.
9. Webster, Gerald M., ed., *Deconvolution*, Geophysics Reprint Series No. 1, Tulsa, Society of Exploration Geophysicists, 1978.
10. Breckenridge, Franklin R., "Acoustic Emission Transducer Calibration by Means of the Seismic Surface Pulse," *Journal of Acoustic Emission*, vol. 1, no. 2, 1982, pp. 87-94.
11. Proctor, Thomas M., Jr., "An Improved Piezoelectric Acoustic Emission Transducer," *JASA*, vol. 71, no. 5, 1982, pp. 1163-1168.
12. Breckenridge, Franklin R., and Greenspan, Martin, "Surface-wave Displacement: Absolute Measurements Using a Capacitive Transducer," *JASA*, vol. 69, no. 4, 1981, pp. 1177-1185.
13. Pollock, A. A., "Acoustic Emission: A Review of Recent Progress and Technical Aspects," *Acoustics and Vibration Progress*, vol. 1, (Stephens, R.W.B., and Leventhall, H.G., eds.) New York, Halsted Press, 1974.
14. Sachse, Wolfgang, and Hsu, Nelson N., "Ultrasonic Transducers for Materials Testing and Their Characterization," *Physical Acoustics*, vol. 14, 1979, pp. 277-406.
15. Bentley, P.G., "A Review of Acoustic Emission for Pressurized Water Reactor Applications," *NDT International*, 1981, pp. 329-335.
16. Breckenridge, Franklin R., Tschiegg, Carl E., and Greenspan, Martin, "Acoustic Emission: Some Applications of Lamb's Problem," *JASA*, vol. 57, no. 3, 1975, pp. 626-631.
17. Lanchon-Magnin, I., Fleischmann, P., Rouby, D., and Goutte, R., "Application of Signal Processing to Acoustic Emission Signals Emitted by Steel Specimens During Tensile Tests," *Ultrasonics*, January 1982, pp. 18-24.
18. Eisenblatter, J., "The Origin of Acoustic Emission—Mechanisms and Models," *Acoustic Emission* (A. R. Nicoll, trans.) Oberursel, Deutsche Gesellschaft fur Metallkunde, 1980.
19. Ceranoglu, A.N., and Pao, Yih-Hsing, "Propagation of Elastic Pulses and Acoustic Emission in a Plate," (3 parts) *Trans. ASME J. Appl. Mech.*, vol. 48, 1981, pp. 125-147, 1981.
20. Love, A. E. H., *A Treatise on the Mathematical Theory of Elasticity*, 4th ed., New York, Dover Publications, 1944.

21. Mindlin, R. D., "Waves and Vibrations in Isotropic Elastic Plates," reprinted from *Structural Mechanics*, Oxford, Pergamon Press, 1960, pp. 199-232.
22. Weaver, R. L., and Pao, Y.-H., "Axisymmetric Elastic Waves Excited by a Point Source in a Plate," ASME Paper 82-WA/APM-15, 1982.
23. Pao, Yih-Hsing, and Gajewski, Ralph R., "The Generalized Ray Theory and Transient Responses of Layered Elastic Solids," *Physical Acoustics*, vol. 13, 1977, pp 184-265.
24. Cagniard, L., *Reflection and Refraction of Progressive Seismic Waves* (E.A. Flinn and C.H. Dix, trans.), New York, McGraw-Hill, 1962.
25. Pekeris, C. L., "The Seismic Surface Pulse," *Geophysics*, vol. 41, 1955, pp. 469-480.

END

FILMED

304

DTIC